Machine Learning
https://doi.org/10.1007/s10994-018-5737-x

Lowest probability mass neighbour algorithms: relaxing the metric constraint in distance-based neighbourhood algorithms

Kai Ming Ting[1] · Ye Zhu[2] · Mark Carman[3] · Yue Zhu[4] · Takashi Washio[5] · Zhi-Hua Zhou[4]

Received: 23 December 2016 / Accepted: 22 June 2018

Abstract
The use of distance metrics such as the Euclidean or Manhattan distance for nearest neighbour algorithms allows for interpretation as a geometric model, and it has been widely assumed that the metric axioms are a necessary condition for many data mining tasks. We show that this assumption can in fact be an impediment to producing effective models. We propose to use mass-based dissimilarity, which employs estimates of the probability mass to measure dissimilarity, to replace the distance metric. This substitution effectively converts nearest neighbour (NN) algorithms into lowest probability mass neighbour (LMN) algorithms. Both types of algorithms employ exactly the same algorithmic procedures, except for the substitution of the dissimilarity measure. We show that LMN algorithms overcome key shortcomings of NN algorithms in classification and clustering tasks. Unlike existing generalised data independent metrics (e.g., quasi-metric, meta-metric, semi-metric, peri-metric) and data dependent metrics, the proposed mass-based dissimilarity is unique because its self-dissimilarity is data dependent and non-constant.

Keywords Nearest neighbour · Distance metric · Lowest probability mass neighbour · Mass-based dissimilarity · Classification · Clustering

1 Introduction and motivation

Many machine learning algorithms rely on a distance measure to provide the closest match between a test point and example points from a database in order to find its nearest neighbours. The distance calculation is the core operation that has been applied to many data mining and machine learning tasks, including density estimation, clustering, anomaly detection and classification.

Editor: Jesse Davis.

✉ Ye Zhu
ye.zhu@ieee.org

Extended author information available on the last page of the article

Published online: 02 July 2018

Despite its widespread applications, research in psychology has pointed out since the 1970's that distance measures do not possess a key property of dissimilarity as judged by humans (Krumhansl 1978; Tversky 1977), namely that two points in a dense region of the space are less similar to each other than two points of the same interpoint distance in a sparse region. Researchers have suggested that a *data dependent dissimilarity* provides better performance than data independent geometric model based distance measures in psychological tests (Krumhansl 1978). For example, two Caucasians will be judged as less similar when compared in Europe (where there are many Caucasians) than in East Asia (where there are few Caucasians and many East Asians).

We introduce a new generic data dependent measure called mass-based dissimilarity which has the above-mentioned characteristic, and we provide concrete evidence that it is a better measure than standard distance measures for two existing algorithms which rely on distance calculations to perform classification and clustering tasks.

The new measure violates two metric axioms, namely, the constancy and minimality of self-dissimilarity. We show that the data dependency has two components: data dependent dissimilarity between identical points (aka self-dissimilarity) and data dependent dissimilarity between two non-identical points. Though there exist data dependent measures which are metrics or pseudo-metrics, they are data dependent on the second component only. The data dependent self-dissimilarity is a unique feature of mass-based dissimilarity.

Mass-based dissimilarity uses an estimate of the probability mass rather than distance as the means to find the closest match neighbourhood. This heralds a fundamental change of perspective: the neighbourhood is now determined by the lowest probability mass neighbours rather than the nearest neighbours.

Simply replacing the distance measure with mass-based dissimilarity effectively converts nearest neighbour (NN) algorithms to Lowest Probability Mass Neighbour (LMN) algorithms. Both types of algorithms employ exactly the same algorithmic procedures, except for the substitution of the dissimilarity measure.

This paper makes the following contributions:

1. Providing a generic data dependent dissimilarity, named mass-based dissimilarity, in which its data dependency has two components: self-dissimilarity and dissimilarity for two non-identical points $\mathbf{x} \neq \mathbf{y}$.
2. Analysing the conditions under which mass-based dissimilarity will overcome key shortcomings of distance-based neighbourhood methods in classification and clustering tasks.
3. Through simulations and empirical evaluation, demonstrating that LMN algorithms overcome key shortcomings of NN algorithms in classification and clustering tasks. This is achieved through the replacement of the distance measure with the mass-based dissimilarity in existing algorithms.
4. Investigating the similarities and differences with existing data dependent measures.

The remainder of the paper is organised as follows. Section 2 presents the proposed mass-based dissimilarity. Section 3 describes the lowest probability mass neighbour (kLMN) algorithm and the analyses on the condition under which the kLMN classifier reduces the error rate of the kNN classifier. Section 4 describes how mass-based clustering can be obtained by simply replacing distance measure with mass-based dissimilarity in an existing density-based clustering algorithm. It also provides the analyses on the condition under which mass-based clustering performs better than density-based clustering. Section 5 describes the shortcomings of existing distance-based neighbourhood methods. Section 6 presents the empirical evaluation results. Related data dependent dissimilarities and metric axioms are discussed in Sect. 7. The relation to distance metric learning, data dependent kernel and similarity-based learning is presented in Sect. 8. A discussion of other issues and conclusions are provided in the last two sections.

2 Mass-based dissimilarity

Geometric model based measures depend solely on the geometric positions of data points to derive distance measures. Instead, mass-based dissimilarity measures mainly depend on the distribution of the data.

The intuition underlying the proposed measures is that the dissimilarity between two points primarily depends on the amount of probability mass of a region of a space covering the two points. Specifically, two points in a dense region are less similar to each other than two points of the same interpoint distance in a sparse region.

In order to turn the intuition above into a useful measure, we need to (a) define what the region covering two points is; and (b) provide a method for estimating its probability mass. Since we do not wish to make any parametric assumptions about the form of the underlying probability distribution that generated the data, we turn to non-parametric tree-based partitioning techniques (described in Sect. 2.2) to define a hierarchy of regions and calculate their probabilities. We can then define a distribution sensitive dissimilarity measure as the probability mass[1] of the smallest region in the hierarchy covering both points. We name the proposed measure: **mass-based dissimilarity**.

We note that the new measure still makes use of a geometric model in order to define a region which encloses neighbouring points. However, once the regions are defined, the dissimilarity between any two points is determined by the probability mass of the smallest region covering the two points. We now formalise the concepts as follows.

Let H denote a hierarchical model that partitions the space $\mathbb{R}^q$ into a set of non-overlapping axis-aligned regions that collectively span $\mathbb{R}^q$. Each internal node (representing a region) in the hierarchy corresponds to the union of its child nodes/regions. Let $\mathcal{H}(D)$ denote the set of all such hierarchical partitions H that are admissible under a data set D such that each non-overlapping region contains at least one point from D. As a result, each hierarchy $H \in \mathcal{H}(D)$ has a finite height and a finite number of external nodes, both have the maximum of $|D|$.

The smallest region covering two points is defined as follows:

Definition 1 $R(\mathbf{x}, \mathbf{y}|H)$, the smallest local region covering $\mathbf{x} \in \mathbb{R}^q$ and $\mathbf{y} \in \mathbb{R}^q$ with respect to the hierarchical partitioning model H of $\mathbb{R}^q$, is defined as:

$$R(\mathbf{x}, \mathbf{y}|H) = \underset{r \in H \ s.t. \{\mathbf{x},\mathbf{y}\} \in r}{\operatorname{argmax}} \ depth(r; H) \tag{1}$$

where $depth(r; H)$ is the depth of the node r in the hierarchical model H.

Note that the probability of data falling into the smallest region containing both $\mathbf{x}$ and $\mathbf{y}$, denoted $P(R(\mathbf{x}, \mathbf{y}|H))$, is analogous to the shortest distance between $\mathbf{x}$ and $\mathbf{y}$ used in the geometric model.

Let $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\mathbf{x}_i \in \mathbf{R}^d$ be a dataset sampled from an unknown probability density function $\mathbf{x}_i \sim F$.

Definition 2 Mass-based dissimilarity of $\mathbf{x}$ and $\mathbf{y}$ wrt D and F is defined as the expectation, over all possible partitionings of the data, of the probability that a randomly chosen point

[1] We use the terms: probability mass or mass or probability, interchangeably hereafter.

$\mathbf{z} \sim F$ would lie in the region $R(\mathbf{x}, \mathbf{y}|H)$:

$$m(\mathbf{x}, \mathbf{y}|D, F) = E_{\mathcal{H}(D)}[P_F(R(\mathbf{x}, \mathbf{y}|H))] \tag{2}$$

where $P_F(\cdot)$ is the probability wrt F; and the expectation is taken over the probability distribution on all hierarchical partitioning $H \in \mathcal{H}(D)$ of the dataset D.

In practice, the mass-based dissimilarity would be estimated from a finite number of models $H_i \in \mathcal{H}(D), i = 1, \ldots, t$ as follows:

$$m_e(\mathbf{x}, \mathbf{y}|D) = \frac{1}{t} \sum_{i=1}^{t} \tilde{P}(R(\mathbf{x}, \mathbf{y}|H_i; D)) \tag{3}$$

where $\tilde{P}(R) = \frac{1}{|D|} \sum_{\mathbf{z} \in D} \mathbb{1}(\mathbf{z} \in R)$ estimates the probability of the region R using the count of data points in that region; and $\mathbb{1}(\cdot)$ is an indicator function.

Hereafter we drop D from the notations when the context is clear.

2.1 Self-dissimilarity

A unique feature of the mass-based dissimilarity is self-dissimilarity, whereby the dissimilarity between a point and itself $m_e(\mathbf{x}, \mathbf{x})$ is non-constant across the space $\mathbb{R}^q$, and ranges over $[0, 1]$ with value depending on the data distribution and the partitioning strategy used. This relaxes the metric axioms which require self-dissimilarity to be minimum and constant.

The non-constancy stems from the fact that the partitions in H can contain different number of data points and thus different probability mass estimates. The non-constancy of self-dissimilarity implies that it cannot be adjusted by simply subtracting a constant from all values. This is unlike self-dissimilarity of other measures which usually take a constant value equal to the minimum dissimilarity.

The differences between m_e and ℓ_p are shown in Table 1. The reasons for the two properties of m_e are given below:

- $\forall \mathbf{x}\mathbf{y} \;\; (\mathbf{x} \neq \mathbf{y}) \rightarrow (m_e(\mathbf{x}, \mathbf{x}) \leq m_e(\mathbf{x}, \mathbf{y}))$

  This is true because $R(\mathbf{x}, \mathbf{x}|H) \subseteq R(\mathbf{x}, \mathbf{y}|H)$. The self-dissimilarity is the base value in which the dissimilarity $m_e(\mathbf{x}, \mathbf{y})$ between two points is measured, i.e., $m_e(\mathbf{x}, \mathbf{y}) \geq m_e(\mathbf{x}, \mathbf{x}) + m_e(\mathbf{y}, \mathbf{y})$. Thus the self-dissimilarity has a direct impact on $m_e(\mathbf{x}, \mathbf{y})$ even though no duplicate points may exist in the dataset.

- $\forall \mathbf{x}\mathbf{y}\mathbf{z} \;\; (\mathbf{y} \neq \mathbf{z}) \nrightarrow (m_e(\mathbf{x}, \mathbf{x}) \leq m_e(\mathbf{y}, \mathbf{z}))$

  This is because the mass distribution for $m_e(\mathbf{x}, \mathbf{x})$ is not constant[2] over the space. If $\mathbf{x}$ is in the region that includes the maximum mass point, then its mass value will be larger than the dissimilarity of some points $\mathbf{y}$ and $\mathbf{z}$ which are close to the fringe or have the minimum mass values. Both mass distributions shown in Fig. 1b, c provide such an example: $m_e(\mathbf{z}, \mathbf{y})$ at either fringes of the distribution is less than the maximum mass $m_e(\mathbf{x}, \mathbf{x})$. This property always holds when the half-space partitioning strategy is used because mass distribution is concave (see below).

[2] Note that mass distribution is not uniform even for a uniform density distribution. See Chen et al. (2015); Ting et al. (2010) for details.

Table 1 m_e versus ℓ_p (the usual metric based on p-norm)

	$\partial(\mathbf{x},\mathbf{x})$	$\partial(\mathbf{x},\mathbf{y}), \forall \mathbf{x} \neq \mathbf{y}$	Properties
m_e	(0,1] DD	(0,1] DD	$\forall \mathbf{xy}\ \ (\mathbf{x} \neq \mathbf{y}) \rightarrow (m_e(\mathbf{x},\mathbf{x}) \leq m_e(\mathbf{x},\mathbf{y}))$ $\forall \mathbf{xyz}\ \ (\mathbf{y} \neq \mathbf{z}) \nrightarrow (m_e(\mathbf{x},\mathbf{x}) \leq m_e(\mathbf{y},\mathbf{z}))$
ℓ_p	0 DI	(0,1] DI	$\forall \mathbf{x};\ \mathbf{y} \neq \mathbf{z},\ \ell_p(\mathbf{x},\mathbf{x}) < \ell_p(\mathbf{z},\mathbf{y})$

$\partial(\cdot,\cdot)$ is a dissimilarity function. DD and DI stand for data dependent and data independent, respectively

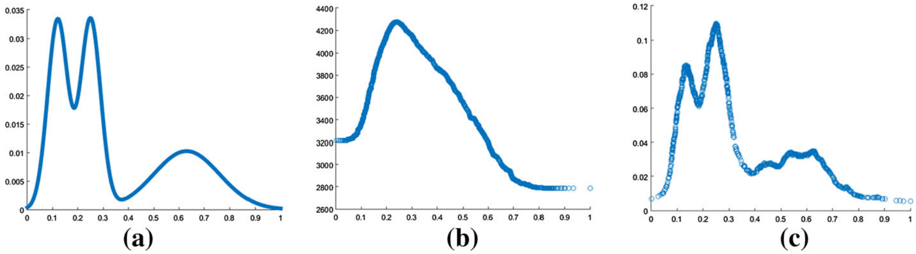


Fig. 1 **a** A true density distribution; **b** $m_e(x, x|D)$ based on half-space and **c** $m_e(x, x|D)$ based on random trees (level = 8) are two mass distributions generated using two different partitioning strategies from a dataset D with 1500 points randomly sampled from (**a**)

In general, the distribution of the data dependent self-dissimilarity is equivalent to the mass distribution defined by Ting et al. (2010), and its properties depend on the hierarchical partitioning strategy $\mathcal{H}$:

- If a **half-space partitioning strategy** is used to define the regions in H, then $m_e(\mathbf{x},\mathbf{x})$ reduces to the half-space mass defined in Chen et al. (2015). Here the mass distribution is always concave within the area bounded by the data, irrespective of the density distribution of the given data set. It has a unique maximum mass point which can be treated as the median of the distribution. The minimum mass values are at the fringes of the distribution (Chen et al. 2015). An example is shown in Fig. 1b.
- If instead a **random tree partitioning strategy** is used to define H, then $m_e(\mathbf{x},\mathbf{x})$ reduces to level-h mass estimation as defined by Ting et al. (2013b). An example is shown in Fig. 1c.

This paper uses the random trees partitioning strategy rather than half-space because the former provides finer granularity in mass-based dissimilarity. The relationships between these two partitioning strategies, and between mass estimation and mass-based dissimilarity are provided in Sect. 2.4.

An intuitive example is provided as follows. An East Asian Asian may be considered more similar to themselves in Europe (where there are fewer East Asians) than they are in East Asia (where there are a lot of East Asians), though they have exactly the same physical features, regardless of whether they are in East Asia or Europe. This is also reflected in Fig. 1c where high (low) density regions have high (low) self-dissimilarity. This self similarity influences the similarity measurement of two different persons. A consequence of the data dependency of self similarity leads to the outcome that two East Asians are more similar in Europe than they are in East Asia.

This example shows that the shortest distance (i.e., zero distance) is not equivalent to the closest (judged) similarity or smallest probability mass; and data dependent self-dissimilarity is an important aspect of a mass-based measure akin to judged dissimilarity.

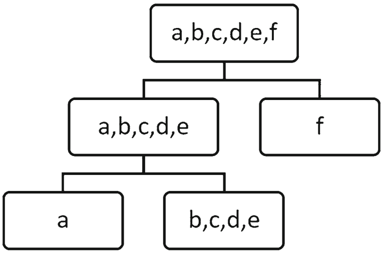


Fig. 2 Example: six points partitioned by a 2-level $iTree$

2.2 Hierarchical partitioning method used to define regions

There are many methods to implement a model to define regions for mass estimation (Ting et al. 2013b). In this paper, we employ a method based on completely random trees[3] to implement mass-based dissimilarity.

We use a recursive partitioning scheme called $iForest$ (isolation Forest) (Liu et al. 2008) to define regions. Though $iForest$ was initially designed for anomaly detection (Liu et al. 2008), it has been shown that it is a special case of mass estimation (Ting et al. 2013b), which will be discussed in Sect. 2.4.

The implementation can be divided into two steps. There are two input parameters to this procedure. They are: t—the number of $iTrees$ (isolation Trees); and ψ—the sub-sampling size used to build each $iTree$. The height limit h of each $iTree$ is automatically set by $\psi : h = \lceil log_2 \psi \rceil$.

The first step is to build an $iForest$ consisting of t $iTrees$ as the partitioning structure R. Each $iTree$ is built independently using a subset $\mathcal{D} \subset D$, sampled without replacement from D, where $|\mathcal{D}| = \psi$. A randomly selected split is employed at each internal node of an $iTree$ to partition the sample set at the node into two non-empty subsets until every point is isolated or the maximum tree height h is reached. Unless stated otherwise, axis-parallel splits are used at each node of an $iTree$ to build an $iForest$. The details of the $iTree$ building process can be found in "Appendix A".

Figure 2 provides an example of six points partitioned by an $iTree$: $m_e(a, b) = 5$, and $m_e(b, f) = 6$, $m_e(d, d) = 4$ and $m_e(f, f) = 1$ (ignoring the normalising term $|\mathcal{D}| = 6$).

After an $iForest$ is built, all points in D are traversed through each tree in order to record the mass of each node.

The second step is the evaluation step. Test points $\mathbf{x}$ and $\mathbf{y}$ are passed through each $iTree$ to find the mass of the deepest node containing both $\mathbf{x}$ and $\mathbf{y}$, i.e., $\sum_i |R(\mathbf{x}, \mathbf{y}|H_i)|$. Finally, $m_e(\mathbf{x}, \mathbf{y})$ is the mean of these mass values over t $iTrees$ as defined below:

$$m_e(\mathbf{x}, \mathbf{y}) = \frac{1}{t} \sum_{i=1}^{t} \frac{|R(\mathbf{x}, \mathbf{y}|H_i)|}{|D|} \tag{4}$$

[3] Note that the random trees are not RandomForests (Breiman 2001) because the trees are unsupervised and completely random, i.e., built without class labels and any attribute selection criterion.

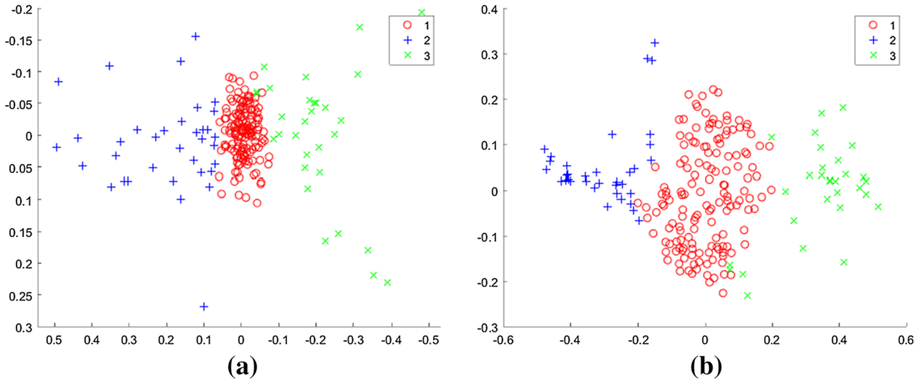


Fig. 3 MDS plots on the Thyroid dataset with two different dissimilarity measures. **a** MDS using distance. **b** MDS using m_e

2.3 Visualising the effects of a data dependent dissimilarity measure

The visualisation in Fig. 3 provides a different perspective of the advantage of using a data dependent dissimilarity measure. Here we performed multidimensional scaling (MDS)[4] with mass-based dissimilarity and distance measures. The red dense region in Fig. 3a becomes sparser and the two sparse regions in Fig. 3a become denser in the MDS plots using the mass-based dissimilarity measure, shown in Fig. 3b. This modified distribution enables all clusters to be detected by an existing clustering algorithm that would not have succeeded otherwise (see Sect. 4.2.2 for details).

2.4 Mass estimation and mass-based dissimilarity

Mass-based dissimilarity is a direct descendant of another line of research, called mass estimation (Ting et al. 2010, 2013b; Chen et al. 2015). Data distribution is often modelled in terms of density distribution. Mass estimation offers an alternative way to model data distribution in terms of mass distribution. Before the advent of mass estimation, $iForest$ (Liu et al. 2008) was initially created for the sole purpose of anomaly detection. Its effectiveness in discerning and ranking anomalous points in a dataset enables it to be used for other tasks which require ranking. Indeed, the use of $iForest$ has been adapted in a content-based information retrieval system called $ReFeat$ (Zhou et al. 2012). The system finds points in a database which is relevant to a query, coupling with relevance feedback. $iForest$ was then recognised as an implementation of mass estimation (Ting et al. 2013b), where the path length (as the anomaly score) traversed by a test point along each $iTree$ is recognised as a proxy to mass. A direct use of mass facilitates the improved versions of $iForest$ and $ReFeat$ (Aryal et al. 2014b).

From another perspective, a special case of mass estimation, called half-space mass (Chen et al. 2015), shares common characteristics of data depth (Liu et al. 1999; Mosler 2013) which is aimed to find the median in a multi-dimensional space. They both model data distribution in terms of center-outward ranking rather than density or linear ranking. They have the following

[4] Multidimensional scaling is a technique for visualising the information contained in a dissimilarity matrix (Borg et al. 2012). An MDS algorithm aims to place each data point in a w-dimensional space ($w = 2$ is used here), while preserving as much as possible the pairwise dissimilarities between them.

characteristics: (i) the resultant mass distribution is concave (convex for data depth) regardless of the underlying density distribution, (ii) the maximum mass point or the minimum data depth point can be considered as a generalisation of the median.

The half-space mass (Chen et al. 2015) is a generalisation of the univariate mass estimation (Ting et al. 2010, 2013b) to multi-dimensional spaces; and it facilitates the extension of the mass estimation of one point to a dissimilarity measure of two points based on mass. The generic definition of mass-based dissimilarity, presented in this paper, encompasses m_p-dissimilarity (Aryal et al. 2014a) as its special case. The details of m_p-dissimilarity are provided in Sect. 7.1.

SNN (dis)similarity (Jarvis and Patrick 1973) can be viewed as an early form of mass-based dissimilarity. The μ-neighbourhood function introduced here is a new form of mass estimator which is defined based on mass-based dissimilarity. Using this function as a template, we can now see that the neighbourhood function based on SNN is an early variant of the μ-neighbourhood function. The details of SNN similarity and its relation to mass-based dissimilarity are provided in Sect. 4.4.

It is interesting to examine the versatility of mass estimation. Thus far, mass estimation has been implemented using trees [including $iForest$ (Liu et al. 2008) and its variants (Ting and Wells 2010; Tan et al. 2011; Ting et al. 2013a)], half-space (Chen et al. 2015) and nearest neighbour (Wells et al. 2014). These implementations have been applied to anomaly detection, clustering, information retrieval, classification, emerging new classes problems in data streams (Mu et al. 2017), and even density estimation based on mass called DEMass (Ting et al. 2013a; Wells et al. 2014).

Given the strong connection between mass estimation and mass-based dissimilarity, the above implementations of mass estimation can potentially be channelled for use in mass-based dissimilarity—$iForest$ is an example used in this paper.

3 Lowest probability mass neighbour classifiers

We now describe the lowest probability mass neighbour (LMN) algorithm which is formed by simply replacing the distance measure in nearest neighbour (NN) algorithm with the mass-based dissimilarity. We focus on classification here. The nearest neighbour and lowest probability mass neighbour for NN and LMN classifiers, respectively, are obtained as follows:

$$NN(\mathbf{x}; D) = \underset{\mathbf{y}\in D}{\operatorname{argmin}}\ \ell_p(\mathbf{x}, \mathbf{y})$$

$$LMN(\mathbf{x}; D) = \underset{\mathbf{y}\in D}{\operatorname{argmin}}\ m_e(\mathbf{x}, \mathbf{y})$$

In this paper, we assume that Euclidean distance is used for NN classifiers, i.e., $p = 2$ for ℓ_p.

The shortcoming of the classifier is given in the first subsection; and the condition under which kLMN will have a lower error rate than kNN is provided in the second subsection. Simulations to demonstrate the analytical results are presented in the third subsection.

3.1 Shortcoming of the kNN classifier

Let $\Gamma \subseteq \mathbb{R}^q$ be a q-dimensional real space which is also a metric space (Γ, ℓ_p) and a probability space $(\Gamma, 2^\Gamma, P)$. Let $\mathcal{X}$ be a subset of Γ and have a finite volume for a classification problem. It is partitioned into two non-overlapping open subsets $\mathcal{X}_T$ and $\mathcal{X}_S$, where $\mathcal{X}_T$ con-

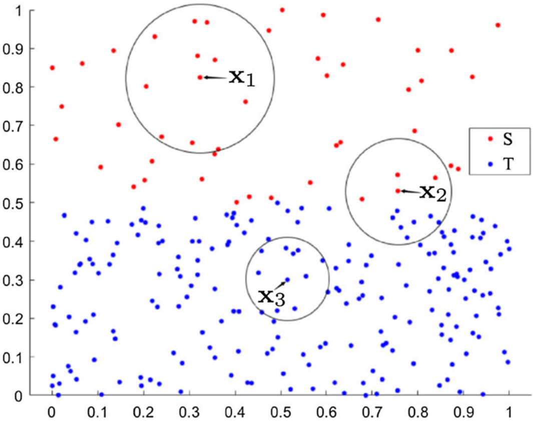


Fig. 4 A dataset has two clusters with different densities. The points in three circles are k nearest neighbours of $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, respectively ($k = 10$). $\mathbf{x}_2$ is a border point with $k_S(\mathbf{x}_2) = 3$ and $k_T(\mathbf{x}_2) = 7$, while $\mathbf{x}_1$ and $\mathbf{x}_3$ are non-border points

tains positive instances only and $\mathcal{X}_S$ contains negative instances only, such that the problem is deterministic with Bayes error rate = 0.

Furthermore, let the probability density of instances be non-zero everywhere in $\mathcal{X}$; and zero outside $\mathcal{X}$, *i.e.*, $P(\mathbf{x}) > 0\ \forall \mathbf{x} \in \mathcal{X}$ and $P(\mathbf{x}) = 0\ \forall \mathbf{x} \in \Gamma \backslash \mathcal{X}$. Assume $\mathcal{X}$ is contiguous and therefore no 'border region' with zero probability can exist between the two classes. Let the density of region $\mathcal{X}_T$ be higher than that of region $\mathcal{X}_S$: $\forall \mathbf{x} \in \mathcal{X}_T$, $\forall \mathbf{y} \in \mathcal{X}_S$, $P(\mathbf{x}) > P(\mathbf{y})$.

Let a training dataset D be a union of a positive instance set D_T and a negative instance set D_S belonging to the dense subset $\mathcal{X}_T$ and the sparse subset $\mathcal{X}_S$, respectively. We consider a k nearest neighbours (kNN) classifier having $k \ll |D|$ with D. Let the set of k nearest neighbours of $\mathbf{x}$ be $NN_k(\mathbf{x})$ where the numbers of instances belonging to $\mathcal{X}_T$ and $\mathcal{X}_S$ are $k_T(\mathbf{x})$ and $k_S(\mathbf{x})$, respectively; and $k_T(\mathbf{x}) + k_S(\mathbf{x}) = k$. For $\mathbf{x} \in \mathcal{X}$, let $B(\mathbf{x})$ be a ball centred at $\mathbf{x}$ with the radius being the k-th nearest neighbour distance from $\mathbf{x}$ in D.

Assume that the curvature of the border between $\mathcal{X}_T$ and $\mathcal{X}_S$ is sufficiently small. Hence, we presume that the border section of $\mathcal{X}_T$ and $\mathcal{X}_S$ covered by any $B(\mathbf{x})$ is effectively linear or straight, when $B(\mathbf{x})$ covers the border. We also assume that the densities of instances in $\mathcal{X}_T$ and $\mathcal{X}_S$ vary smoothly; and thus presume that the densities of instances in the intersections of any $B(\mathbf{x})$ with $\mathcal{X}_T$ and $\mathcal{X}_S$ are almost uniform, and they are denoted as $\rho_T(\mathbf{x})$ and $\rho_S(\mathbf{x})$, respectively. These assumptions mostly hold when D_T and D_S are massive and smoothly distributed, and k nearest neighbour distance of $\mathbf{x}$ is very small because of $k \ll |D|$.

A kNN border point $\mathbf{x} \in D_S$ is defined as one which has $NN_k(\mathbf{x})$ such that $k_T(\mathbf{x}) \geq 1$; and a kNN non-border point $\mathbf{x} \in D_S$ is one which has $NN_k(\mathbf{x})$ such that $k_T(\mathbf{x}) = 0$. Similarly, a kNN border point and a non-border point $\mathbf{x} \in D_T$ are defined to have $NN_k(\mathbf{x})$ such that $k_S(\mathbf{x}) \geq 1$ and $k_S(\mathbf{x}) = 0$, respectively. Figure 4 illustrates some example border and non-border points in a dataset. Here, we further introduce expectations of $\rho_T(\mathbf{x})$ and $\rho_S(\mathbf{x})$ over the instances which are the kNN border points in D and denote them as $\bar{\rho}_T$ and $\bar{\rho}_S$, respectively. Then, the following theorem holds.

Theorem 1 *In a dataset consisting of a dense subset ($\mathcal{X}_T$) and a sparse subset ($\mathcal{X}_S$) which do not overlap, the kNN classifier's misclassification rate in $\mathcal{X}_S$: ε_S is a probabilistically increasing function of the density ratio $\bar{\rho}_T / \bar{\rho}_S$, and that in $\mathcal{X}_T$: ε_T is most probably zero. The majority of the misclassification errors occurs in the region of the sparse $\mathcal{X}_S$, bordering the dense $\mathcal{X}_T$.*

The proof is given in "Appendix B".

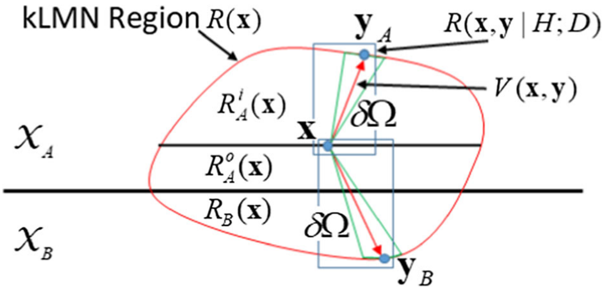


Fig. 5 Two regions $\mathcal{X}_A$ and $\mathcal{X}_B$ have uniform but different densities $\rho_A(\mathbf{x}) \neq \rho_B(\mathbf{x})$. $R(\mathbf{x})$ is a kLMN region of a border point $\mathbf{x} \in \mathcal{X}_A$ and thus intersecting with both $\mathcal{X}_A$ and $\mathcal{X}_B$. $R(\mathbf{x}, \mathbf{y}|H; D)$, shown as a rectangle because of axis-parallel partitions, is a region defining $m_e(\mathbf{x}, \mathbf{y})$. $V(\mathbf{x}, \mathbf{y})$ is the intersection of $R(\mathbf{x})$ with a solid angle cone of $\mathbf{x}$ capturing $\mathbf{y}$. $R^i_A(\mathbf{x})$ and $R^o_A(\mathbf{x})$ are sub-regions of $R(\mathbf{x}) \cap \mathcal{X}_A$ partitioned by a line crossing $\mathbf{x}$ and parallel to the border between $\mathcal{X}_A$ and $\mathcal{X}_B$. $R^i_A(\mathbf{x})$ and $R^o_A(\mathbf{x})$ are farther from and closer to the border, respectively. The sub-region $R_B(\mathbf{X})$ is $R(\mathbf{x}) \cap \mathcal{X}_B$

3.2 The kLMN classifier has smaller error rate than the kNN classifier under certain condition

Under the identical problem setting with the kNN classifier, we consider a k lowest probability mass neighbours (kLMN) classifier having k ($\ll |D|$) that built from the dataset D and the mass-based dissimilarity. The mass-based dissimilarity $m_e(\mathbf{x}, \mathbf{y})$ between $\mathbf{x}$ and $\mathbf{y}$ is given by the expected data mass in a region $R(\mathbf{x}, \mathbf{y}|H; D)$ enclosing both $\mathbf{x}$ and $\mathbf{y}$ and containing the smallest number of data points in D within a hierarchical partition model H, where axis-parallel partitions are generated randomly. Because the region $\mathcal{X}$ is finite, $R(\mathbf{x}, \mathbf{y}|H; D)$ is effectively finite even if the partitions include some open regions. Therefore, we assume a finite $R(\mathbf{x}, \mathbf{y}|H; D)$ without loss of generality.

Let $LMN_k(\mathbf{x})$ be the set of k lowest probability mass neighbours of $\mathbf{x}$, and let $m_e(\mathbf{x})$ be the mass-based dissimilarity between $\mathbf{x}$ and the k-th lowest probability mass neighbour of $\mathbf{x}$, *i.e.*, $m_e(\mathbf{x}) = \max_{\mathbf{y} \in LMN_k(\mathbf{x})} m_e(\mathbf{x}, \mathbf{y})$. Moreover, define a kLMN region $R(\mathbf{x})$ as a set of all points where the mass-based dissimilarity between $\mathbf{x}$ and any point in $R(\mathbf{x})$ is less than or equal to $m_e(\mathbf{x})$, *i.e.*,

$$R(\mathbf{x}) = \{\mathbf{y} \in \mathcal{X} \mid m_e(\mathbf{x}, \mathbf{y}) \leq m_e(\mathbf{x})\}.$$

By definition, $R(\mathbf{x})$ includes all points in $LMN_k(\mathbf{x})$.

Here, we introduce two open subsets of Γ; $\mathcal{X}_A$ and $\mathcal{X}_B$ having the same associated mathematical definitions and assumptions as in $\mathcal{X}_T$ and $\mathcal{X}_S$, specified in Sect. 3.1. Their densities $\rho_A(\mathcal{X})$ and $\rho_B(\mathcal{X})$ are assumed to be different. $\mathcal{X}_A$ and $\mathcal{X}_B$ are interchangeably used to represent $\mathcal{X}_T$ and $\mathcal{X}_S$ in the proof of Theorem 2.

Figure 5 shows a kLMN region $R(\mathbf{x})$ over $\mathcal{X}_A$ and $\mathcal{X}_B$, where $\mathbf{x}$ is a border point in terms of $LMN_k(\mathbf{x})$ and located in $\mathcal{X}_A$. Consider a point $\mathbf{y}$ on the edge of $R(\mathbf{x})$. By the above definition of $R(\mathbf{x})$, any $\mathbf{y}$ on the edge of $R(\mathbf{x})$ has identical $m_e(\mathbf{x}, \mathbf{y}) = m_e(\mathbf{x})$. For example, points $\mathbf{y}_A$ and $\mathbf{y}_B$ in Fig. 5 have dissimilarities $m_e(\mathbf{x}, \mathbf{y}_A)$ and $m_e(\mathbf{x}, \mathbf{y}_B)$, respectively, which are equal to $m_e(\mathbf{x})$. We define sub-regions $R_A(\mathbf{x})$ and $R_B(\mathbf{x})$ which are the intersections of $R(\mathbf{x})$ with $\mathcal{X}_A$ and $\mathcal{X}_B$, respectively. We split $R_A(\mathbf{x})$ into $R^i_A(\mathbf{x})$ and $R^o_A(\mathbf{x})$ by a line crossing $\mathbf{x}$ and parallel to the border between $\mathcal{X}_A$ and $\mathcal{X}_B$, where $R^o_A(\mathbf{x})$ is the sub-region closer to the border; and $R^i_A(\mathbf{x}) = R_A(\mathbf{x}) \setminus R^o_A(\mathbf{x})$. We further introduce $V(\mathbf{x}, \mathbf{y})$: an intersection of $R(\mathbf{x})$

with a cone of $\mathbf{x}$ having a solid angle $\delta\Omega$ which includes point $\mathbf{y}$; and let the probability mass in $V(\mathbf{x},\mathbf{y})$ be $\delta P(V(\mathbf{x},\mathbf{y}))$.

Let $\tilde{E}_{\mathcal{H}(D)}[\bar{\rho}(R(\mathbf{x},\mathbf{y}|H;D))]$ be an expectation of the average density in region $R(\mathbf{x},\mathbf{y}|H;D)$ over $\mathcal{H}(D)$, and let $\bar{\rho}(V(\mathbf{x},\mathbf{y}))$ be the average density in $V(\mathbf{x},\mathbf{y})$, respectively. $\bar{\rho}(V(\mathbf{x},\mathbf{y}))$ and $\tilde{E}_{\mathcal{H}(D)}[\bar{\rho}(R(\mathbf{x},\mathbf{y}|H;D))]$ are considered to be almost identical, and they are in the interval between $\rho_A(\mathbf{x})$ and $\rho_B(\mathbf{x})$, since they are both average densities in some vicinity of $\mathbf{x}$ and $\mathbf{y}$. We further introduce $\phi(\mathbf{x})$ which is an upper bound of a ratio $\{P(R_A^o(\mathbf{x}))+P(R_B(\mathbf{x}))\}/P(R_A^i(\mathbf{x}))$ at $\mathbf{x}$ where $P(R_*(\mathbf{x}))$ is the probability mass in $R_*(\mathbf{x})$, and let $\bar{\phi}$ be the average of $\phi(\mathbf{x})$ taken over all the kLMN border points in D. Their rigorous definitions are provided in "Appendix C".

To simplify the analysis of the kLMN classifier, we introduce the concept of 'effective dimensions' of $R(\mathbf{x},\mathbf{y}|H,D)$. Assume that the border section of $\mathcal{X}_T$ and $\mathcal{X}_S$ is effectively linear in $R(\mathbf{x})$, as in the case in Sect. 3.1. Effective dimensions are defined as the dimensions of $R(\mathbf{x},\mathbf{y}|H,D)$ which are non-orthogonal to the normal line of the border between $\mathcal{X}_T$ and $\mathcal{X}_S$. We let the number of the effective dimensions of $R(\mathbf{x},\mathbf{y}|H,D)$ be $\tilde{q}$. For example, $\tilde{q}$ of the hyper-rectangle $R(\mathbf{x},\mathbf{y}|H,D)$ depicted in Fig. 6a is 1, since its horizontal dimension is parallel to the border and thus orthogonal to the normal line. On the other hand, $\tilde{q}$ in Fig. 6b is 2, because both dimensions are non-orthogonal to the normal line. Note that $1 \le \tilde{q} \le q$ holds. Then, we obtain the following theorem.

Theorem 2 *In a dataset consisting of a dense subset ($\mathcal{X}_T$) and a sparse subset ($\mathcal{X}_S$) which do not overlap, assuming $\bar{\rho}(V(\mathbf{x},\mathbf{y})) \simeq \tilde{E}_{\mathcal{H}(D)}[\bar{\rho}(R(\mathbf{x},\mathbf{y}|H,D))]$, the kLMN classifier's misclassification rate in $\mathcal{X}_S$: ε_S and that in $\mathcal{X}_T$: ε_T have the following properties:*

(i) If the effective dimension of the kLMN classifier $\tilde{q}=1$, both ε_S and ε_T are most probably very small.
(ii) If $\tilde{q}>1$, ε_S is most probably a non-negligible and increasing function of the density ratio $(\bar{\rho}_T/\bar{\rho}_S)^{1-1/\tilde{q}}$ when $\bar{\phi}$ is close to the upper bound in the interval $[1,(\bar{\rho}_T/\bar{\rho}_S)^{1-1/\tilde{q}}]$, and
(iii) If $\tilde{q}>1$, ε_T is most probably zero.

The proof is given in "Appendix C".

Theorem 2 indicates that the kLMN classifier shows non-negligible error ε_S which probabilistically increases with $(\bar{\rho}_T/\bar{\rho}_S)^{1-1/\tilde{q}}$; while ε_T is almost zero. It also suggests a possibility to make both ε_S and ε_T very small by controlling the number of the effective dimensions $\tilde{q}$ at unity.

In summary, Theorems 1 and 2 unveil that

(a) The majority of misclassification errors of both the kNN and kLMN classifiers occur in the region of sparse subset, bordering the dense subset. Intuitively, the fact that the sparse subset has significantly fewer points means that it is harder to find points from the sparse subset (than those from the dense subset) to form the k lowest probability mass neighbours (or k nearest neighbours) of a border point. As a result, a border point of either subset is more likely to be predicted to belong to the dense subset. This is the reason why the error rate is higher in the sparse subset than in the dense subset for either kNN or kLMN.
(b) The kLMN classifier has a lower misclassification error than the kNN classifier. This is because the rates of increase of ε_S of the kNN and kLMN classifiers become identical, i.e., $\bar{\rho}_T/\bar{\rho}_S$, only if $\tilde{q}\to\infty$. Otherwise, kLMN's rate of increase of ε_S is slower than that of kNN's by a factor of $(\bar{\rho}_T/\bar{\rho}_S)^{1/\tilde{q}}$ as $(\bar{\rho}_T/\bar{\rho}_S)$ increases. This conclusion can be

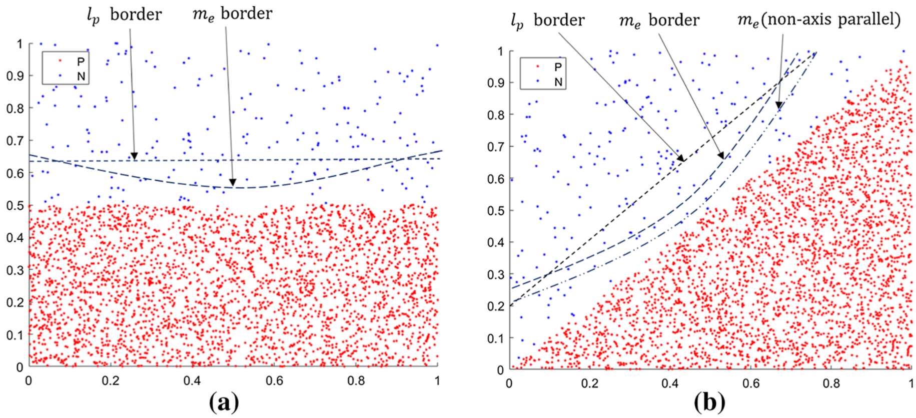


Fig. 6 Data distributions used in two simulations. Label "P" and "N" indicate positive and negative points, respectively. The dash lines indicate the boundaries due to kNN (ℓ_p border) and kLMN (m_e border). An additional boundary due to kLMN, where m_e is implemented using non-axis parallel splits, is also shown. **a** Data distribution of simulation 1. **b** Data distribution of simulation 2

made despite the fact that an analytical comparison of ε_S is intractable between the kNN and kLMN classifiers.

3.3 Simulations

This section provides two simulations to demonstrate the analytical results presented in the last two sections.

We built one dataset for each of the two simulations. Both datasets have two classes, and each class is generated using uniform density distribution. The only difference is that the boundary of the first dataset is axis parallel while the second is non-axis parallel, as shown in Fig. 6a, b.

In each simulation, we gradually increased the density ratio between the two classes in the training set, and then evaluated the performance of the kNN and kLMN classifiers (where $k = \sqrt{n}$; n=training set size). We report the result in terms of false negative rate (FNR), false positive rate (FPR), and error rate (ERR) which is the average value of FNR and FPR, where the positive class is the one having the higher density. The test set consists of 1250 instances for each class. This is to ensure that a sufficient number of points along the entire border in order to obtain the FPR and FNR which can be compared across different density ratios.[5]

Tables 2 and 3 show the results of the two simulations. In agreement with Theorems 1 and 2, the results show that (i) the majority of the errors of the kNN and kLMN classifiers are in the sparse region, bordering the dense region (having high false positive rate); and (ii) kLMN has a lower false positive rate than kNN, and this leads to a lower error rate.

Figure 6 further demonstrates that the boundary of kLMN, unlike that of kNN, is not parallel to the (true) boundary between the dense and sparse regions. Rather, the two boundaries (between the m_e border and the true border) are closer in the centre, and the gap widens

[5] Note that a test set of a fixed size which has the same density ratio as the training set is not used because it does not allow us to compare the error rates across different density ratios—due to the fact that the number of instances in the minority gets diminishing small as the density ratio increases; and the FPR becomes increasing less representative because there are fewer points along the border.

Table 2 Performance of the kNN and kLMN classifiers in simulation 1. Each result is an average over 10 runs

Data size		l_p			m_e		
#pos	#neg	FNR	FPR	ERR	FNR	FPR	ERR
200	200	0.02	0.02	0.02	0.00	0.01	0.01
600	200	0.00	0.08	0.04	0.01	0.00	0.01
1000	200	0.00	0.14	0.07	0.03	0.00	0.02
1400	200	0.00	0.18	0.09	0.02	0.00	0.01
1800	200	0.00	0.20	0.10	0.02	0.00	0.01
2200	200	0.00	0.23	0.11	0.02	0.01	0.01
2600	200	0.00	0.24	0.12	0.01	0.02	0.02
3000	200	0.00	0.26	0.13	0.01	0.05	0.03

Table 3 Performance of the kNN and kLMN classifiers in simulation 2. Average over 10 runs

Data size		l_p			m_e			m_e (non-axis parallel)		
#pos	#neg	FNR	FPR	ERR	FNR	FPR	ERR	FNR	FPR	ERR
200	200	0.04	0.02	0.03	0.06	0.03	0.04	0.04	0.02	0.03
600	200	0.00	0.13	0.07	0.01	0.12	0.06	0.00	0.08	0.04
1000	200	0.00	0.20	0.10	0.00	0.16	0.08	0.00	0.12	0.06
1400	200	0.00	0.23	0.12	0.00	0.19	0.10	0.00	0.14	0.07
1800	200	0.00	0.26	0.13	0.00	0.21	0.10	0.00	0.14	0.07
2200	200	0.00	0.27	0.14	0.00	0.22	0.11	0.00	0.16	0.08
2600	200	0.00	0.30	0.15	0.00	0.24	0.12	0.00	0.19	0.09
3000	200	0.00	0.32	0.16	0.00	0.25	0.13	0.00	0.19	0.10

towards the edges. This is because the mass distribution has higher mass in the centre and lower mass at the edges (Ting et al. 2010), even in a uniform density distribution.

4 Mass-based clustering

Here we describe how density-based clustering can be transformed into mass-based clustering by simply replacing the distance measure with mass-based dissimilarity.

A new neighbourhood function using mass-based dissimilarity is introduced in the first subsection. Its characteristics are described in the second subsection. The condition under which mass-based clustering will provide a better clustering result than density-based clustering is presented in the third subsection. The fourth subsection discusses a closely related similarity measure.

4.1 μ-neighbourhood mass

We introduce a new function: μ-neighbourhood mass which counts the number of points that have mass-based dissimilarity less than or equal to a maximum value μ. This is similar to

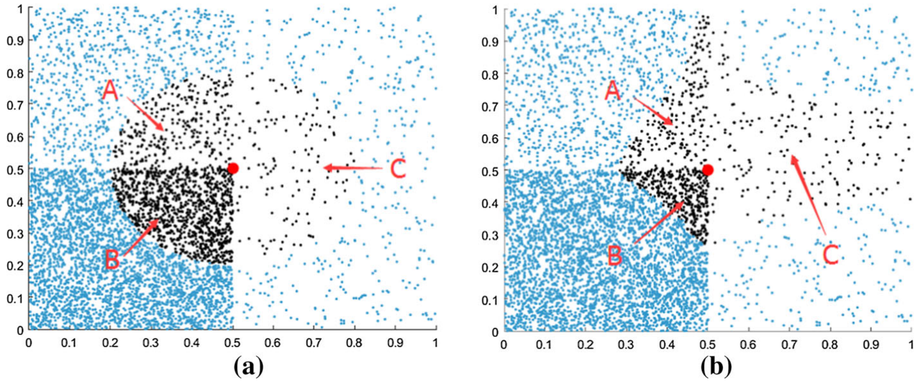


Fig. 7 **a** and **b** show the two sets of points defined by ϵ-neighbourhood density ($\epsilon = 0.25$) and μ-neighbourhood mass (level $= 4$ *iForest* and $\mu = 0.5$), respectively, on a dataset having three areas of different densities, with reference to the red point (0.5, 0.5). The dark-coloured dots denote the set of points defined by either ϵ-neighbourhood or μ-neighbourhood estimator; and the blue-coloured dots denote the set of points outside. **a** Points defined by ϵ-neighbourhood. **b** Points defined by μ-neighbourhood (Color figure online)

ϵ-neighbourhood density[6] which denotes the set of points that has up to a maximum distance ϵ defined by a distance measure. It is defined as follows:

$$M_\mu(\mathbf{x}) = \#\{\mathbf{y} \in D \mid m_e(\mathbf{x}, \mathbf{y}) \leq \mu\}$$

Like standard ϵ-neighbourhood density, μ-neighbourhood mass makes an estimate based on a dissimilarity measure. However, the estimate is defined in terms of the expected probability mass (instead of distance). Like ϵ, the parameter μ controls the set size: a large μ defines a large set; and a small μ defines a small set.

In addition, the model used determines the general shape of $M_\mu(\mathbf{x})$ (e.g., diamond or circle); and the shape is symmetric only if the distribution of self-dissimilarity is symmetric wrt $\mathbf{x}$. The boundary of $M_\mu(\mathbf{x})$ is closer (/further) to $\mathbf{x}$ if the self dissimilarities of points between $\mathbf{x}$ and the boundary have high (/low) mass.

Figure 7a, b compare ϵ-neighbourhood with μ-neighbourhood on a dataset having three areas of different densities. Note that the volume of the region occupied by points of μ-neighbourhood mass depends on the data distribution—it is small in dense areas and large in sparse areas, as demonstrated by areas A, B and C in Fig. 7b. Note that the overall shape is not symmetric. In contrast, the ϵ-neighbourhood forms a region which is independent of the data distribution with constant volume, in addition to regular and symmetric shape.

Figure 8a shows that the region occupied by the set of points of μ-neighbourhood becomes symmetric only in the case of a uniform density distribution. Figure 8 shows that the shape of the region depends on the implementation: axis-parallel and non axis-parallel random trees yield diamond and spherical shapes, respectively. These shapes are similar to those of ℓ_p of $p = 1$ and $p = 2$.

Figure 9 shows the influence of self-dissimilarity on M_μ: For a fixed μ, the μ-neighbourhood $M_\mu(\mathbf{x})$ covers a small region when $\mathbf{x}$ is in the high mass area of self-dissimilarity (and vice versa).

A conceptual comparison between the two neighbourhood functions is given in Table 4. In order to obtain non-zero M_μ, μ must be set higher than $\max_{\mathbf{z} \in D} m_e(\mathbf{z}, \mathbf{z})$.

[6] There are different definitions of density. In this paper, we discuss the density on which DBSCAN (Ester et al. 1996) is based.

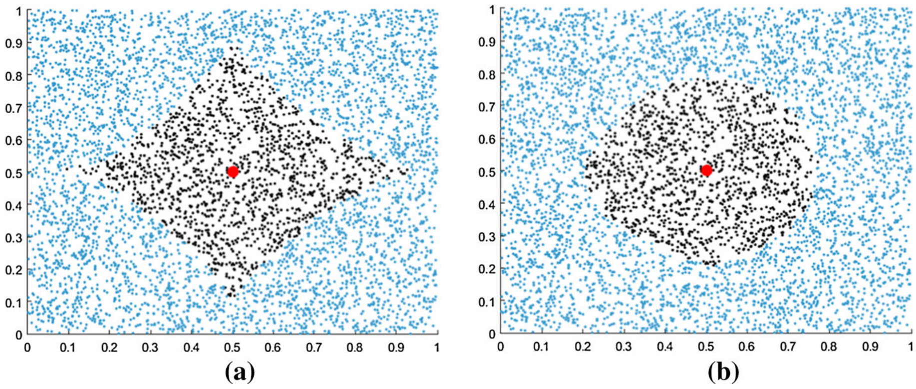


Fig. 8 **a** and **b** show the two sets of points defined by μ-neighbourhood mass ($\mu = 0.55$) using axis-parallel $iForest$ and non axis-parallel $iForest$, respectively, on a dataset with uniform density distribution with reference to the red point (0.5,0.5). **a** μ-neighbourhood (axis-parallel $iForest$). **b** μ-neighbourhood (non axis-parallel $iForest$) (Color figure online)

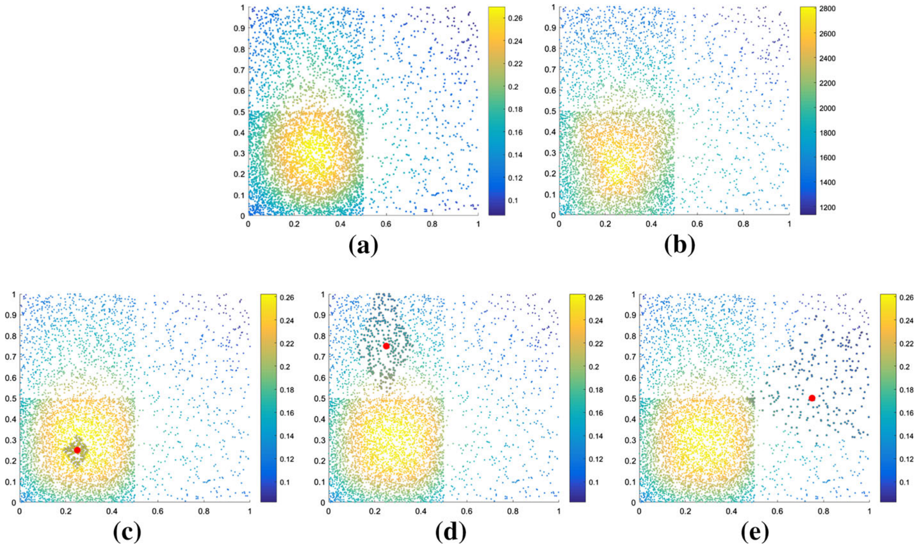


Fig. 9 Self-dissimilarity $m_e(\mathbf{x}, \mathbf{x})$ shown in (**a**) directly influences the measurement $M_\mu(\mathbf{x})$ ($\mu = 0.38$) shown in (**b**). $M_\mu(\mathbf{x})$ ($\mu = 0.60$) at three different points are shown in (**c**), (**d**) and (**e**), where the underlay is the self-dissimilarity. The set of points defined by $M_\mu(\mathbf{x})$ is indicated as boldfaced points. **a** Self-dissimilarity m_e. **b** μ-neighbourhood mass M_μ. **c** M_μ at $\mathbf{x} = (0.25, 0.25)$. **d** M_μ at $\mathbf{x} = (0.25, 0.75)$. **e** M_μ at $\mathbf{x} = (0.75, 0.75)$

For the purpose of clustering, mass can be used in a similar way as density to identify core points, i.e., points having high mass values are core points; and those having low mass values are noise.

The next subsection describes characteristics of μ-neighbourhood mass which make it a better candidate than ϵ-neighbourhood density to identify core points, especially in a dataset which has clusters of varying densities.

Table 4 μ-neighbourhood mass versus ϵ-neighbourhood density

	μ-neighbourhood mass	ϵ-neighbourhood density
Definition	$M_\mu(\mathbf{x}) = \#\{\mathbf{y} \in D \mid m_e(\mathbf{x}, \mathbf{y}) \le \mu\}$ $M_\mu(\mathbf{x}) = 0$ if $m_e(\mathbf{x}, \mathbf{x}) > \mu$ $M_\mu(\mathbf{x}) > 0$ if $m_e(\mathbf{x}, \mathbf{x}) \le \mu$	$N_\epsilon(\mathbf{x}) = \#\{\mathbf{y} \in D \mid \ell_p(\mathbf{x}, \mathbf{y}) \le \epsilon\}$ $\forall \mathbf{x}; 0 < \epsilon \le 1, N_\epsilon(\mathbf{x}) \ge 1$
Region formation	Depend on $m_e(\mathbf{x}, \mathbf{x})$ and data distribution around $\mathbf{x}$	Geometric distance from $\mathbf{x}$
Volume	Depend on the data distribution (small in the dense region and big in the sparse region)	Constant
Shape	Not regular or symmetric	Regular and symmetric

4.2 Characteristics of μ-neighbourhood mass

4.2.1 Rate of change of M_μ

For a distribution of self-dissimilarity $m_e(\mathbf{x}, \mathbf{x})$ which has the same number of modes as that in the density distribution, we have the following observation for the rate of change of $M_\mu(\mathbf{x})$:

Observation 1 *The rate of change of $M_\mu(\mathbf{x})$ (per unit change in μ) is larger in dense regions than in sparse regions.*

While this is the same characteristic as in ϵ-neighbourhood density function, there are two key differences. First, the non-zero M_μ values do not start at the same μ; whereas all non-zero N_ϵ values start (i.e., a value just above zero) at the same ϵ. Second, the rate of change of $M_\mu(\mathbf{x})$ is proportional to self-dissimilarity $m_e(\mathbf{x}, \mathbf{x})$ and its curvature (concave slows and convex accelerates the rate). In contrast, the rate of change of ϵ-neighbourhood density is proportional to the density of $\mathbf{x}$ only.

In addition, while $m_e(\mathbf{x}, \mathbf{x})$ is high in dense regions and low in sparse regions in general, its distribution is centre-outward which is high at the centre and low at the edges (Ting et al. 2010; Chen et al. 2015), even in the case of a uniform density distribution.

Let $\mathbf{x}_{p1}, \mathbf{x}_{p2}$ and $\mathbf{x}_{p3}$ be the points having the largest self-dissimilarity in each of the three clusters in Fig. 10a; and $\mathbf{x}_{v1}$ and $\mathbf{x}_{v2}$ be the points having the smallest self-dissimilarity in each of the two valleys. Figure 10b, c shows the rates of change of M_μ and N_ϵ, respectively, for the five points.

Figure 10b shows that the five points have different starting μ values which produce the first non-zero M_μ; and each starting μ is the self-dissimilarity of the point. For example, the curve for $\mathbf{x}_{v1}$ starts at $\mu = m_e(\mathbf{x}_{v1}, \mathbf{x}_{v1})$ and $M_\mu(\mathbf{x}_{v1}) = 1$; and it starts after $\mathbf{x}_{v2}$ and $\mathbf{x}_{p3}$, and before $\mathbf{x}_{p1}$ and $\mathbf{x}_{p2}$.

The relative rate of change for the five points becomes apparent when the starting μ values are re-aligned to the same point. This is shown in Fig. 10d. Peak 1 and Peak 2 have approximately the same fastest rate and they are the highest two peaks. Peak 3 has the next fastest rate. The convex curvature for each peak increases the rate, in comparison with valleys which have the concave curvature. This is the reason why Peak 3 has a faster rate than Valley 1, though the latter has higher self-dissimilarity. Valley 2 has the lowest self-dissimilarity and rate of change.

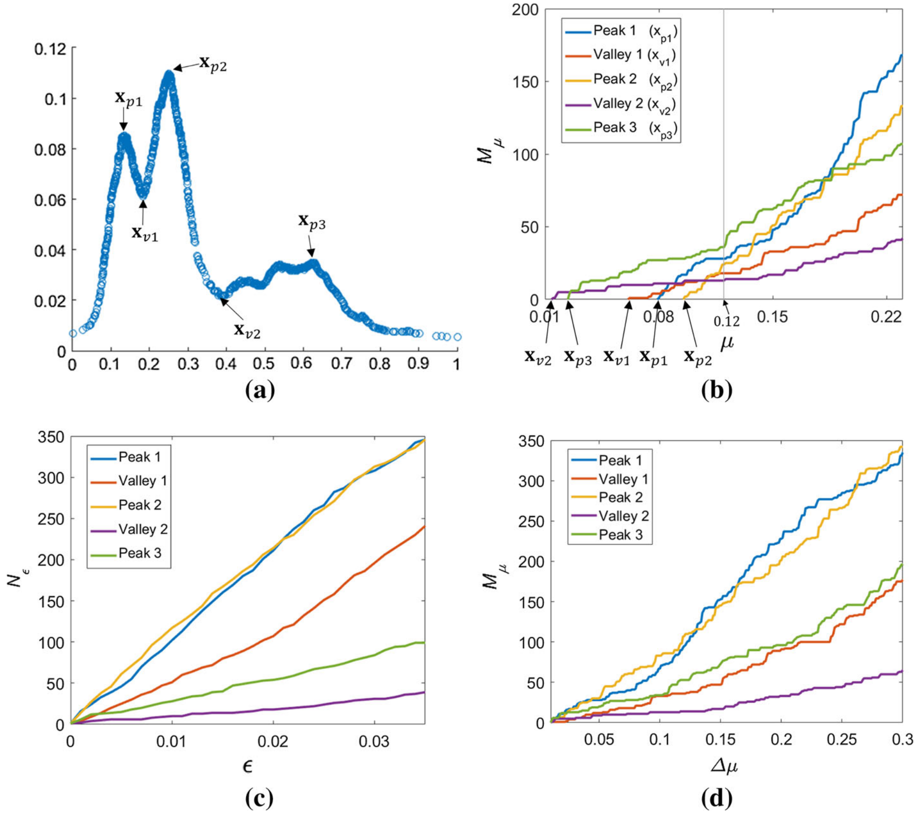


Fig. 10 Rates of change of N_ϵ and M_μ (shown in (**c**) and (**d**), respectively) for the five points in the distribution of $m_e(\mathbf{x}, \mathbf{x})$ shown in (**a**). **b** The starting μ value for each line (which has the first non-zero M_μ) is the self-dissimilarity of the point, e.g., Peak 1 starts with $\mu = m_e(\mathbf{x}_{p1}, \mathbf{x}_{p1})$ and $M_\mu(\mathbf{x}_{p1}) = 1$. **d** This is the same plot as (**b**), except all the staring points are re-aligned to $\mu = 0$ so that the relative rate of change can be compared readily. **a** Self-dissimilarity distribution of Fig. 1a. **b** M_μ versus μ. **c** N_ϵ versus ϵ. **d** M_μ versus $\Delta\mu$ (all starting points aligned)

4.2.2 Converting from inseparable distributions to separable distributions for clustering

Another interesting characteristic of μ-neighbourhood mass is that the valleys of a data distribution can be constrained within a small range of mass value by setting an appropriate μ. This characteristic is especially important in clustering algorithms which rely on a global threshold to identify core points before grouping the core points into separate clusters. Having all the valleys close to a small mass value, a global threshold slightly larger than this value will identify the majority of the core points of all clusters, irrespective of the densities of the clusters.

In contrast, ϵ-neighbourhood density does not possess this characteristic as it estimates density distribution and its valleys can have hugely different densities. As a result, a clustering algorithm, such as DBSCAN (Ester et al. 1996) which employs the ϵ-neighbourhood density estimator and relies on a global threshold to identify core points, is unable to detect all clusters of varying densities. This kind of distribution, shown in Fig. 11a, is called an inseparable

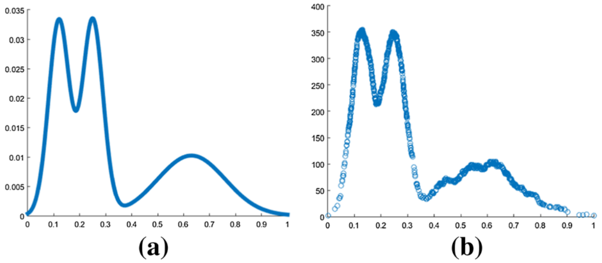


Fig. 11 Density distribution of an "inseparable distribution" and its estimations using ϵ-neighbourhood density ($\epsilon = 0.03$), μ-neighbourhood mass based on $iForest$ (level= 8 and $\mu = 0.2$) from a sample of 1500 points of (**a**). **a** True density distribution. **b** ϵ-neighbourhood density. **c** μ-neighbourhood mass

distribution because it is impossible for DBSCAN (Ester et al. 1996) to identify all the clusters in the distribution using a global threshold.

Possessing the above-mentioned characteristic, the inseparable distribution (in terms of density) will exhibit as a separable distribution (in terms of mass). Examples of ϵ-neighbourhood density and μ-neighbourhood mass are given in Fig. 11b, c, respectively. A single threshold can then be used to identify all three clusters in the reshaped distribution in Fig. 11c. This is impossible if the density distribution is estimated instead, as shown in Fig. 11b.

In summary, μ-neighbourhood mass is able to convert all valleys of hugely different densities to become valleys of low mass of approximately the same value by using an appropriate μ. As a result of the mass-based dissimilarity measure used, the estimated distribution has a characteristic which enables an inseparable distribution in terms of density to be reshaped into a separable distribution in terms of mass for clustering.

4.3 Condition under which mass-based clustering performs better than density-based clustering

The exact conditions under which density-based clustering fails and mass-based clustering succeeds in identifying all clusters in a dataset are given in the following two subsections.

4.3.1 Condition under which density-based clustering fails

Here we reiterate the condition under which density-based clustering fails, recently disclosed by Zhu et al. (2016).

If a density-based clustering algorithm uses a global threshold on the estimated density to identify core points and links neighbouring core points together to form clusters, then the requirement given in Eq. 5, based on the estimated density and the density-based clusters, provides a necessary condition for the algorithm to be able to identify all clusters (Zhu et al. 2016):

$$\min_{k \in \{1,\ldots,\varsigma\}} h_k > \max_{i \neq j \in \{1,\ldots,\varsigma\}} g_{ij} \tag{5}$$

where h_k is the density of the mode of each cluster C_k from a total of ς clusters; and g_{ij} is the largest of the minimum estimated density along any path linking clusters C_i and C_j.

The condition requires that the estimated density at the mode of any cluster is greater than the maximum g_{ij} along any path linking any two modes. It implies that there must exist a threshold τ that can be used to break all paths between the modes by assigning regions with the estimated density less than τ to noise, i.e.,

$$\exists_{\tau} \forall_{k, i \neq j \in \{1,\ldots,\varsigma\}} \; h_k \geqslant \tau > g_{ij}$$

In summary, on a dataset having density distribution of the following condition (Zhu et al. 2016):

$$\min_{k \in \{1,\ldots,\varsigma\}} h_k \not> \max_{i \neq j \in \{1,\ldots,\varsigma\}} g_{ij} \tag{6}$$

the density-based clustering will fail to detect all clusters in the dataset.

4.3.2 Condition under which density-based clustering fails but mass-based clustering succeeds

By using μ-neighbourhood mass function instead of ϵ-neighbourhood density function in DBSCAN, we effectively convert the density-based clustering algorithm to a mass-based clustering algorithm, where clusters are defined in terms of mass rather than density.

Let $\mathbf{c}_k$ be the mode of cluster C_k, $k \in \{1, \ldots, \varsigma\}$ in the distribution of self-dissimilarity mass $m_e(\mathbf{x}, \mathbf{x})$[7]; and when using $M_\mu(\cdot)$, μ must be set more than the maximum value of self-dissimilarity.[8]

Observation 2 *For a density distribution satisfying condition (6), there exist some* $\mu > \max_{k \in \{1,\ldots,\varsigma\}} m_e(\mathbf{c}_k, \mathbf{c}_k)$ *such that the distribution of* $M_\mu(\cdot)$ *satisfies the following condition:*

$$\min_{k \in \{1,\ldots,\varsigma\}} M_\mu(\mathbf{c}_k) > \max_{i \neq j \in \{1,\ldots,\varsigma\}} \acute{\mathfrak{g}}_{ij} \tag{7}$$

where $\acute{\mathfrak{g}}_{ij}$ *is the largest of the minimum estimated* $M_\mu(\cdot)$ *along any path linking cluster* C_i *and* C_j.

The following reasoning relies on Observation 1: the rate of change of $M_\mu(\cdot)$ is faster in dense regions than in sparse regions.

Let $\mathfrak{g}_a$ and $\mathfrak{g}_b$ be the maximum and minimum of $\mathfrak{g}_{ij}$ in the distribution of self-dissimilarity $m_e(\cdot, \cdot)$; and $\min_{k \in \{1,\ldots,\varsigma\}} m_e(\mathbf{c}_k, \mathbf{c}_k) \not> \max_{i \neq j \in \{1,\ldots,\varsigma\}} \mathfrak{g}_{ij}$, as in condition (6) of the density distribution of the same dataset. Let $\mathbf{x}_a$ and $\mathbf{x}_b$ be the points for $\mathfrak{g}_a$ and $\mathfrak{g}_b$, respectively; and $\mathbf{x}_h = \text{argmax}_{\mathbf{c}_k, k \in \{1,\ldots,\varsigma\}} m_e(\mathbf{c}_k, \mathbf{c}_k)$ is the peak of all clusters. In plain language, $\mathbf{x}_a$ and $\mathbf{x}_b$ are valleys in the dense and sparse regions, respectively; and $\mathbf{x}_h$ is the peak of all dense regions. Therefore, the self-dissimilarity has the characteristic: $m_e(\mathbf{x}_h, \mathbf{x}_h) > m_e(\mathbf{x}_a, \mathbf{x}_a) > m_e(\mathbf{x}_b, \mathbf{x}_b)$.

In the distribution of $M_\mu(\cdot)$, the aim is to find μ such that $M_\mu(\mathbf{x}_a) \approx M_\mu(\mathbf{x}_b)$ and $\min_{k \in \{1,\ldots,\varsigma\}} M_\mu(\mathbf{c}_k) > max(M_\mu(\mathbf{x}_a), M_\mu(\mathbf{x}_b))$. That is, condition (7) will be satisfied because every mode will have mass more than all valleys.

Let $\rho(M_\mu)$ be the rate of increase of $M_\mu(\cdot)$ for per unit increase in μ. For $\mu > m_e(\mathbf{x}_a, \mathbf{x}_a)$, $\rho(M_\mu(\mathbf{x}_a)) > \rho(M_\mu(\mathbf{x}_b))$ based on Observation 1 and that $\mathbf{x}_a$ is in a denser region than $\mathbf{x}_b$.

[7] The modes can be defined based on $M_\mu(\cdot)$; but the definition based on $m_e(\mathbf{x}, \mathbf{x})$ is simpler which does not require an additional parameter μ. Also note that the modes defined based on mass may or may not be the same as those defined by density. This does not affect the clusters discovered since the linking process (to form a cluster) does not rely on the modes.

[8] This is another example of the influence of self-dissimilarity. This is because $M_\mu(\mathbf{x}) = 0$ for $m_e(\mathbf{x}, \mathbf{x}) > \mu$. If point $\mathbf{c}$ has the maximum self-dissimilarity, μ shall be set to some multiple of $m_e(\mathbf{c}, \mathbf{c})$ in order to get the maximum mass of $M_\mu(\cdot)$.

Note that $\mu = m_e(\mathbf{x}_a, \mathbf{x}_a)$ will give $M_\mu(\mathbf{x}_a) = 1$ and $M_\mu(\mathbf{x}_b) > 1$ because $m_e(\mathbf{x}_a, \mathbf{x}_a) > m_e(\mathbf{x}_b, \mathbf{x}_b)$. Because $M_\mu(\cdot)$ is a monotonic increasing function of μ; and $M_\mu(\mathbf{x}_a)$ begins with a low base but at a faster rate, $M_\mu(\mathbf{x}_a)$ will catch up with $M_\mu(\mathbf{x}_b)$ at some $\mu > m_e(\mathbf{x}_a, \mathbf{x}_a)$.

Because $m_e(\mathbf{x}_h, \mathbf{x}_h) > m_e(\mathbf{x}_a, \mathbf{x}_a)$, $\mu = m_e(\mathbf{x}_h, \mathbf{x}_h)$ will give $M_\mu(\mathbf{x}_h) = 1$ and $M_\mu(\mathbf{x}_a) > 1$. A value of $\mu > m_e(\mathbf{x}_h, \mathbf{x}_h)$ shall be set such that $\min_{k\in\{1,\ldots,\varsigma\}} M_\mu(\mathbf{c}_k) > max(M_\mu(\mathbf{x}_a), M_\mu(\mathbf{x}_b))$. An example of these increases can be found in Fig. 10b, where $\mu \geq 0.12$ satisfy this requirement.

The above increase in μ assumes that the increases in $M_\mu(\mathbf{x}_a)$ and $M_\mu(\mathbf{c}_k)$ occur in the dense region and the increase in $M_\mu(\mathbf{x}_b)$ occurs in the sparse region such that the relative rate of change between dense and sparse regions stays approximately the same.

Thus, the above aim can be achieved by increasing μ surpassing $m_e(\mathbf{x}_h, \mathbf{x}_h)$, provided the relative rate of increase of M_μ satisfies the stated assumption. □

Satisfying the above condition, the mass-based clustering algorithm employing a threshold τ can be used to identify all clusters, as it breaks all paths between the modes by assigning regions with estimated $M_\mu(\cdot)$ less than τ to noise, i.e.,

$$\exists_\tau \forall_{k,i\neq j\in\{1,\ldots,\varsigma\}} M_\mu(\mathbf{c}_k) \geqslant \tau > \acute{\mathfrak{g}}_{ij}$$

Is it possible that μ-neighbourhood mass may convert a separable (density) distribution to an inseparable (mass) distribution? We are unaware of any such conditions. We believe that such conditions are rare in practice because the parameter μ provides sufficient flexibility to morph from an inseparable distribution to a separable distribution. Such conditions did not appear in our experiments reported in Sect. 6.2.

4.3.3 Simulations

Figure 12 shows simulations of two inseparable distributions in which density-based clustering fails to identify all clusters in each case; and the conversion to separable distributions using μ-neighbourhood mass where mass-based clustering successfully identifies all clusters in each case. This is despite the fact that both density-based clustering and mass-based clustering are using exactly the same clustering procedure, except for the substitution of the dissimilarity measure.

4.4 Relation to *SNN* similarity

A measure based on shared nearest neighbours (SNN) in k nearest neighbours has been proposed for clustering (Jarvis and Patrick 1973):

> "Data points are similar to the extent that they share the same nearest neighbours; in particular, two data points are similar to the extent that their respective k nearest neighbour lists match. In addition, for this similarity measure to be valid, it is required that the tested points themselves belong to the common neighbourhood."

SNN (dis)similarity has been used to replace the distance measure in DBSCAN as a way to overcome its inability to find all clusters of varying densities (Ertöz et al. 2003).

Here we show that SNN similarity is a mass-based similarity, and the corresponding neighbourhood estimator is a variant of μ-neighbourhood mass estimator.

Using a similar notation, let $R(\mathbf{x}|H)$ be the (implicit) region which covers the k nearest neighbours of $\mathbf{x}$, where H is kNN. $SNN(\mathbf{x}, \mathbf{y})$ can be defined as the neighbourhood mass

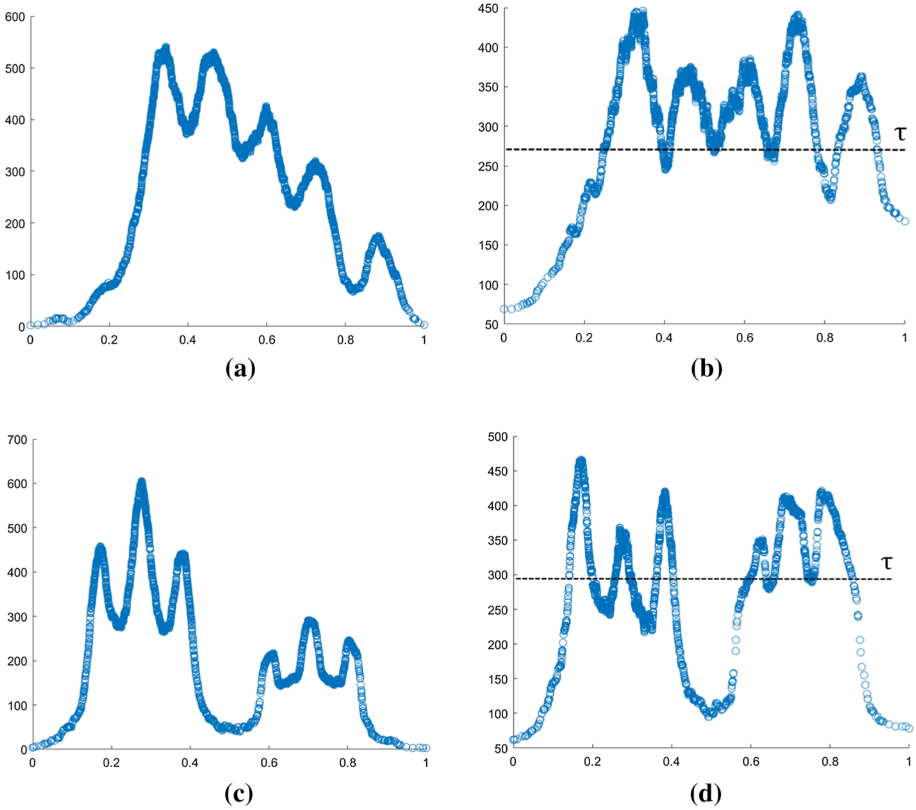


Fig. 12 **a** and **c** are the distributions of ϵ-neighbourhood density of two "inseparable distribution" datasets. **b** and **d** are the distributions of the μ-neighbourhood mass of (**a**) and (**c**) based on $iForest$ (level= 8 and $\mu = 0.22$), respectively. **a** ϵ-neighbourhood density. **b** μ-neighbourhood mass of (**a**). **c** ϵ-neighbourhood density. **d** μ-neighbourhood mass of (**c**)

(i.e., the number of nearest neighbours) shared by $\mathbf{x}$ and $\mathbf{y}$:

$$SNN(\mathbf{x}, \mathbf{y}) = |R(\mathbf{x}|H) \cap R(\mathbf{y}|H)| \tag{8}$$

where the intersection must include both $\mathbf{x}$ and $\mathbf{y}$; otherwise $SNN(\mathbf{x}, \mathbf{y}) = 0$.

Let $s_k(\mathbf{x}, \mathbf{y}) = SNN(\mathbf{x}, \mathbf{y})/k$. The neighbourhood function based on the SNN similarity can be expressed as:

$$M_\sigma(\mathbf{x}) = \#\{\mathbf{y} \in D \mid s_k(\mathbf{x}, \mathbf{y}) \geq \sigma\}$$

Note that $M_\sigma(\mathbf{x})$, like $M_\mu(\mathbf{x})$, cannot be treated as a density because the volume used to compute M_σ for every $\mathbf{x}$ is not constant, given a fixed σ. In other words, we disclose that SNN clustering algorithm (Ertöz et al. 2003) is a type of a mass-based clustering method, not a density-based clustering method. This is despite the fact that SNN employs the DBSCAN procedure by replacing the distance measure with the (inverse) SNN similarity (Ertöz et al. 2003; Tan et al. 2005).

The advantages of using $iForest$ instead of k nearest neighbours to estimate the neighbourhood mass are:

Table 5 Key functions and key shortcomings of algorithms that rely on distance in four tasks and their replacement functions due to mass-based dissimilarity

Algorithm	Key function	Key shortcoming	Replacement function
kNN classifier (kNN)	Find nearest neighbours	Poor classification predictions in cases where classes have varying densities	Find lowest probability mass neighbours
Multi-label kNN classifier (MLkNN)	Estimate class-conditioned likelihoods using a frequency estimate based on k nearest neighbours	Poor likelihood estimation in cases where the local neighbourhood covers regions of varying densities	Estimate class-conditioned likelihoods using a frequency estimate based on k lowest probability mass neighbours
Density-based clustering	Identify core points which have high density using distance-based neighbourhood estimation	Inability to find all clusters of varying densities	Identify core points which have high probability mass using probability mass-based neighbourhood estimation
kNN anomaly detector	Identify anomalies as points with the longest distance to the kth nearest neighbours	Inability to detect local anomalies (due to clusters of varying densities)	Identify anomalies as points having the highest probability mass of the kth lowest probability mass neighbours

- The SNN similarity matrix is sensitive to the parameter of neighbourhood list size k. In contrast, $iForest$ works well with a default setting.
- The SNN clustering algorithm (Ertöz et al. 2003) has $O(k^2n^2)$ time complexity or $O(n^3)$ when $k = \sqrt{n}$ or larger and $n = |D|$. Yet, the same algorithm using mass-based dissimilarity has the same $O(n^2)$ time complexity as DBSCAN, except an additional preprocessing to compute the dissimilarity matrix which takes $O(t \log \psi(\psi + n^2))$ or $O(n^2)$ since $\psi \ll n$. When distance measure is used, the cost for the dissimilarity matrix is also $O(n^2)$.

5 Shortcomings of existing distance-based algorithms

The neighbourhood of a point has been used for different functions in various data mining tasks. Table 5 summarises the key functions and key shortcomings of four existing algorithms relying on the distance measure. It is instructive to see that a mere replacement of distance with the mass-based dissimilarity in these algorithms changes the perspective from finding the nearest neighbourhood to the lowest probability mass neighbourhood, though both denote the least dissimilar neighbourhood. The corresponding 'new' functions are described in the last column of Table 5. Although the rest of the procedures in each algorithm are unchanged, the mass-based dissimilarity overcomes the key shortcomings of these algorithms through finding the lowest probability mass neighbourhood, rather than the nearest neighbourhood.

Table 6 Properties of datasets used in the experiments

Dataset	Data size	#Dimensions	#Classes
Air	359	64	3
Australian	690	14	2
Breast	277	9	2
Corel	10, 000	67	100
German	1000	24	2
Heart	270	13	2
Ionosphere	351	35	2
Vote	435	16	2
Vowel	528	10	11
WBC	683	9	2
Forest	523	27	4
ILPD	579	9	2
Messidor	1151	19	2
Parkinson	1040	26	2
Wilt	500	5	2
GPS	163	6	2
Phishing	11, 055	31	2
QSAR	1055	41	2
Spam	4601	57	2
Urban	168	147	9
Libras	360	90	15
Thyroid	215	5	3
Segment	2310	19	7
WDBC	569	30	2
Wine	178	13	3
Pendig	10, 992	16	10
Seeds	210	7	3
Iris	150	4	3
S1	900	2	3
S2	1500	2	3

As our analyses are focused on classification and clustering, the empirical evaluations in the next section are conducted on these two tasks. The results of the evaluations in multi-label classification and anomaly detection are presented by Ting et al. (2016).

6 Empirical evaluation

The evaluations in classification and clustering are provided in the following two subsections; and the runtime evaluation is presented in the third subsection.

Thirty datasets are used in the experiments, where twenty eight are from the UCI Machine Learning Repository (Lichman 2013) and two are synthetic datasets (S1 and S2). Table 6 presents the properties of these datasets. We have focused on datasets containing numeric

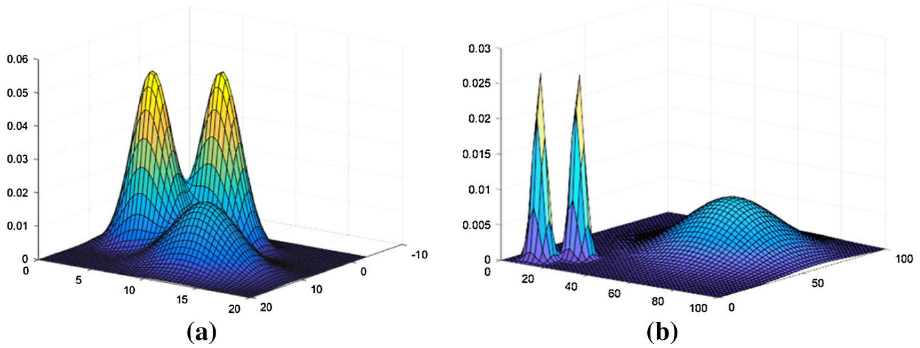


Fig. 13 Density distributions of S1 and S2. **a** S1: "inseparable distribution". **b** S2: "separable distribution"

attributes only where Euclidean distance has the natural interpretation. The majority of the first twenty datasets are 2-class problems which are the focus of the analysis for classification. Therefore, they are used for evaluating kNN and kLMN classifiers. More than half of the last twenty datatsets have the number of classes more than 2. They are chosen to examine the capability of clustering algorithms. All these datasets represent a good mix of different numbers of attributes (2–147), data sizes (150–11055) and classes (2–100).

All datasets are normalised using the *min-max* normalisation (unless stated otherwise) to yield each attribute to be in [0,1] before the experiments begin.

S1 and S2 are the synthetic datasets created to examine the condition under which density-based clustering fails (Zhu et al. 2016), stated in Sect. 4.3. S1 is an "inseparable distribution" which contains 3 Gaussian clusters $N(mean, std)$ with means located at (x_1, x_2)=(3.3, 9.3), (8, 5), (12, 12), and standard deviations $std = 3, 3, 8$ in each dimension; and each cluster has 300 points. S2 is a "separable distribution" which has 3 Gaussian clusters $N(mean, std)$ with means located at $(x_1, x_2) = (10, 10), (20, 20), (60, 60)$, and $std = 2, 2, 11$ in each dimension; and each cluster has 500 points. The density plots of S1 and S2 are shown in Fig. 13.

6.1 kNN classifiers versus kLMN classifiers

This section compares the kNN classifier with the kLMN classifier to assess the relative utility of distance measure and mass-based dissimilarity.

In addition, we also include the Extended Nearest Neighbour classifier (ENN) (Tang and He 2015) and two supervised approaches to distance metric learning called large margin nearest neighbour (LMNN) (Weinberger and Saul 2009) and geometric mean metric learning (GMML) (Zadeh et al. 2016) in the comparison.

ENN is an improved version of kNN which makes its prediction based on not only the nearest neighbors of each test point, but also those instances which have the test point as their nearest neighbors.

LMNN learns a distance metric such that k nearest neighbours (of a test point) belong to the same class, and points from different classes are separated by a large margin from the k nearest neighbours. Because LMNN uses the maximum margin in their objective function, there is no closed form solution.

GMML is a recent method for distance metric learning. It learns a linear transformation (distance metric) such that instances of different classes are more dissimilar and those of the same class are more similar. The linear transformation is Mahalanobis distance parameterized by a symmetric positive definite matrix which has a close-form solution.

A conceptual comparison between distance metric learning and mass-based dissimilarity is provided in Sect. 8. Here we denote LMNN as the kNN classifier which employs the LMNN learned metric; and GMML as the kNN classifier which employs the GMML learned metric.

All classifiers used in the experiments are implemented in MatLab.[9]

Parameters used are given as follows. The number of nearest neighbours k is set to 5 for kNN, ENN, LMNN, GMML and kLMN. We set the number of dimensions of the learned metric in LMNN and GMML as the original number of dimensions on each dataset. The default setting (i.e., $\psi = 256$ and $t = 100$) (Liu et al. 2008) is used to generate $iForest$, which is used to compute m_e for kLMN.

We report the average accuracy of 5-fold cross-validation on each dataset. A post-hoc Nemenyi test[10] is used to examine whether the difference between any two classifiers is significant.

The result shown in Table 7 (on normalised datasets) demonstrates that kLMN has the best result, having the best accuracy on 7 datasets; followed by GMML, LMNN, ENN and kNN on 5, 3, 3 and 2 datasets, respectively. kLMN has the highest average rank among the five algorithms shown in Fig. 14a. This result also shows that mass-based dissimilarity provides a significantly better closest match neighbourhood than the distance measure used for classification in kNN. Although KLMN is not significantly better than LMNN and GMML, kLMN is still the preferred choice because it does not need a computationally expensive learning process.

The result on the unnormalised data shown in Table 8 discloses an interesting phenomenon: normalisation has the highest impact on kNN and ENN, where huge differences can be seen on Heart, Parkinson, Wilt, GPS, SPAM and Urban (comparing their individual results between Tables 7 and 8). The sum of the absolute difference between accuracies with and without normalisation for each of the five algorithms is shown in Table 9. The impact is smaller with LMNN and GMML; but still huge differences can be seen on Austra (LMNN only), Parkinson (GMML only) and Urban. The smallest impact is with kLMN, where there are no huge differences on all datasets. This shows that kLMN has the highest capability to deal with varying data scales in different attributes of the same dataset; followed by LMNN and GMML.

The source of this capability in kLMN is $iForest$, where the random split in each node of a tree almost renders null and void the effect of the value range of each attribute while maintaining the relative order of the two subsets (i.e., one subset has larger values than the other) as a result of the split. In evaluating a test point, its value on a certain attribute influences which of the two branches of a node it traverses. In other words, the outcome of mass-based dissimilarity is invariant to linear scaling of an attribute.[11] Thus, the value ranges between attributes can differ hugely but they do not influence the outcome of the dissimilarity measure

[9] ENN is obtained from http://www.ele.uri.edu/faculty/he/research/ENN/ENN.html; LMNN is obtained from https://bitbucket.org/mlcircus/lmnn; GMML is obtained from https://github.com/PouriaZ/GMML; whereas kLMN, $iForest$ and kNN are our implementations.

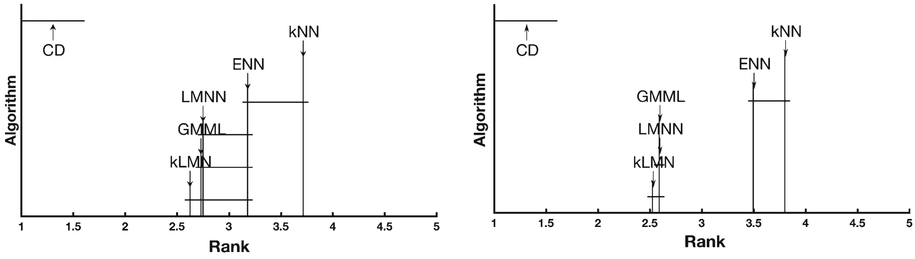
[10] This test (conducted after the Friedman test) (Demšar 2006) is conducted to examine whether the performance difference between any two algorithms is significant. The algorithms under comparison were ranked on each dataset according to their accuracy, where the best one is rank 1 and so on. Then, the post-hoc Nemenyi test is used to calculate the critical difference value (CD) for each algorithm.

[11] Note that kLMN's results in Tables 7 and 8 would be identical if a deterministic algorithm is used to generate $iForest$. The differences we see are a result of the randomisation process used in $iForest$.

Table 7 A comparison of kNN, ENN, LMNN, GMML and kLMN on *data with min-max normalisation*

Dataset	kNN	ENN	LMNN	GMML	kLMN
Air	0.938 ± 0.007	**0.958 ± 0.006**	0.955±0.011	0.953±0.023	0.941 ± 0.019
Austra	0.854 ± 0.008	0.855 ± 0.005	0.845 ± 0.009	0.832±0.008	**0.861 ± 0.011**
Breast	0.665 ± 0.016	0.593 ± 0.035	0.684±0.009	**0.757 ± 0.017**	0.680 ± 0.026
Corel	0.216 ± 0.003	0.250 ± 0.006	0.229 ± 0.002	0.256 ± 0.004	**0.279 ± 0.006**
German	0.702 ± 0.015	0.692 ± 0.008	0.701 ± 0.017	0.694 ± 0.013	**0.718 ± 0.019**
Heart	**0.844 ± 0.024**	0.819 ± 0.023	0.830 ± 0.022	0.811 ± 0.021	0.841 ± 0.030
Ionosphere	0.817 ± 0.005	0.863 ± 0.006	0.874 ± 0.005	0.856 ± 0.014	**0.889 ± 0.005**
Vote	0.931 ± 0.010	**0.947 ± 0.017**	0.940±0.007	0.943±0.010	0.926 ± 0.017
Vowel	0.865 ± 0.018	**0.920 ± 0.016**	0.876±0.015	0.894±0.007	0.870 ± 0.024
Wbc	0.974 ± 0.009	0.969 ± 0.008	0.968 ± 0.010	0.966 ± 0.004	**0.975 ± 0.008**
Forest	0.887 ± 0.010	0.883 ± 0.010	0.883 ± 0.013	**0.888 ± 0.005**	0.885 ± 0.011
ILPD	**0.684 ± 0.009**	0.591 ± 0.019	0.679 ± 0.009	0.629 ± 0.030	0.678 ± 0.017
Messidor	0.630 ± 0.016	0.643 ± 0.003	0.661 ± 0.010	**0.661 ± 0.012**	0.652 ± 0.019
Parkinson	0.993 ± 0.002	0.996 ± 0.002	**1.000 ± 0.000**	0.995 ± 0.002	0.965 ± 0.018
Wilt	0.953 ± 0.003	0.974 ± 0.005	0.980 ± 0.001	**0.982 ± 0.001**	0.972 ± 0.002
GPS	0.838 ± 0.047	0.800 ± 0.019	0.844 ± 0.047	0.800 ± 0.008	**0.844 ± 0.044**
QSAR	0.845 ± 0.003	0.853 ± 0.010	**0.859 ± 0.008**	0.851 ± 0.008	0.858 ± 0.005
Phishing	0.879 ± 0.007	0.895 ± 0.008	0.902 ± 0.010	**0.922 ± 0.006**	0.894 ± 0.007
Spam	0.896 ± 0.007	0.908 ± 0.003	**0.919 ± 0.010**	0.900 ± 0.003	0.916 ± 0.004
Urban	0.806 ± 0.023	0.842 ± 0.029	0.800 ± 0.033	0.836 ± 0.030	**0.861 ± 0.025**
#Top 1	2	3	3	5	7

Results of 5-fold CV: average accuracy and standard deviation. A boldfaced figure indicates that the algorithm has the best accuracy on a dataset. The last row shows the number of datasets in which each algorithm has achieved the best accuracy



(a)

(b)

Fig. 14 Critical difference (CD) diagram of the post-hoc Nemenyi test ($\alpha = 0.05$). The difference between two algorithms is significant if the gap between their ranks is larger than the CD. A line connecting two algorithms indicates that the rank gap between them is smaller than the CD. **a** Normalised datasets showed in Table 7. **b** Unnormalised datasets showed in Table 8

Table 8 A comparison of kNN, ENN, LMNN, GMML and kLMN on *data without normalisation*

Dataset	kNN	ENN	LMNN	GMML	kLMN
Air	0.811 ± 0.025	0.862 ± 0.015	0.914 ± 0.033	0.935 ± 0.016	**0.949 ± 0.021**
Austra	0.657 ± 0.021	0.688 ± 0.025	0.714 ± 0.015	**0.843 ± 0.008**	0.838 ± 0.011
Breast	0.704 ± 0.037	0.647 ± 0.017	0.720 ± 0.026	0.738 ± 0.020	**0.744 ± 0.037**
Corel	0.181 ± 0.005	0.228 ± 0.002	0.192 ± 0.006	0.230 ± 0.003	**0.239 ± 0.004**
German	0.681 ± 0.009	0.631 ± 0.015	0.711 ± 0.008	0.701 ± 0.024	**0.725 ± 0.017**
Heart	0.648 ± 0.043	0.641 ± 0.014	0.726 ± 0.030	**0.826 ± 0.014**	0.815 ± 0.019
Ionosphere	0.791 ± 0.013	0.863 ± 0.026	0.846 ± 0.014	0.860 ± 0.023	**0.880 ± 0.015**
Vote	0.924 ± 0.010	0.922 ± 0.011	**0.949 ± 0.012**	0.929 ± 0.013	0.938 ± 0.007
Vowel	0.844 ± 0.014	**0.945 ± 0.004**	0.804 ± 0.018	0.897 ± 0.008	0.854 ± 0.016
Wbc	0.970 ± 0.004	0.971 ± 0.007	0.971 ± 0.002	0.972 ± 0.005	**0.974 ± 0.003**
Forest	0.883 ± 0.014	0.887 ± 0.019	**0.887 ± 0.018**	0.862 ± 0.022	0.884 ± 0.016
ILPD	0.653 ± 0.018	0.628 ± 0.024	0.657 ± 0.016	**0.702 ± 0.023**	0.655 ± 0.025
Messidor	0.633 ± 0.013	0.670 ± 0.009	0.652 ± 0.005	**0.683 ± 0.015**	0.660 ± 0.007
Parkinson	0.867 ± 0.004	0.860 ± 0.012	**0.996 ± 0.001**	0.853 ± 0.083	0.969 ± 0.011
Wilt	0.970 ± 0.001	0.979 ± 0.002	0.987 ± 0.002	**0.987 ± 0.002**	0.970 ± 0.001
GPS	0.719 ± 0.017	0.738 ± 0.019	0.812 ± 0.010	0.775 ± 0.012	**0.850 ± 0.021**
QSAR	0.817 ± 0.010	0.816 ± 0.016	0.845 ± 0.013	0.853 ± 0.009	**0.858 ± 0.010**
Phishing	0.882 ± 0.007	0.898 ± 0.009	0.904 ± 0.009	**0.924 ± 0.009**	0.894 ± 0.007
Spam	0.784 ± 0.007	0.813 ± 0.005	0.872 ± 0.007	0.862 ± 0.013	**0.925 ± 0.003**
Urban	0.382 ± 0.028	0.406 ± 0.042	0.552 ± 0.067	0.345 ± 0.068	**0.812 ± 0.040**
#Top 1	0	1	3	6	10

Results of 5-fold CV: average accuracy and standard deviation

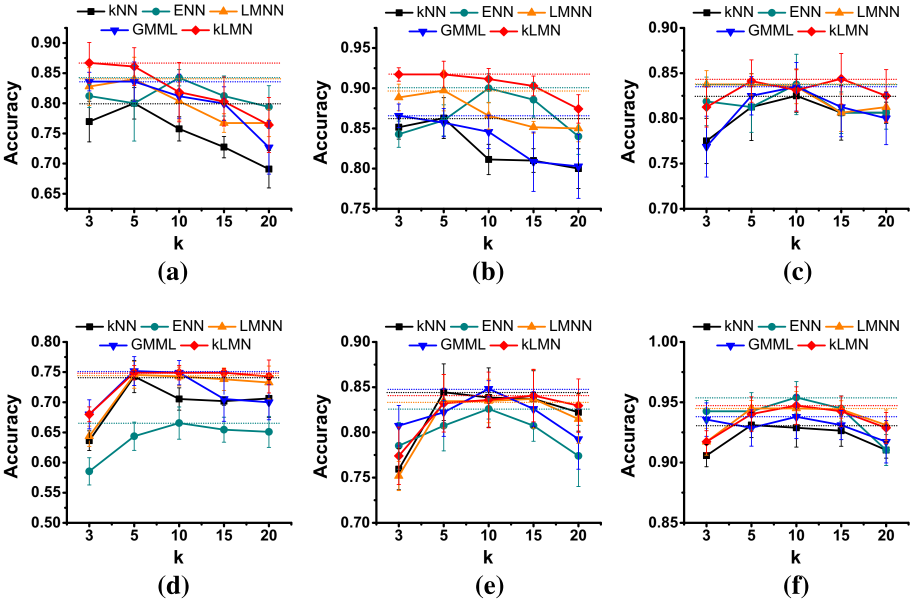
Table 9 The absolute difference between the accuracies with and without normalization in each dataset showed in Tables 7 and 8. The result showed is the sum over the 20 datasets

	kNN	ENN	LMNN	GMML	kLMN
Sum of absolute difference	1.540	1.472	0.860	0.949	0.308

substantially; unlike in the case of ℓ_p. This is a bonus advantage of mass-based dissimilarity over distance measures.

Although both LMNN and GMML have almost identical ranking in both Fig. 14a, b, they have large differences in some datasets, e.g., Breast, Corel, ILPD, GPS and Spam on the normalised datasets. Both LMNN and GMML are significantly better than kNN on the normalised datasets, and better than both kNN and ENN on the unnormalised datasets. We also found that there is no significant difference between kNN and ENN on both the normalised and unnormalised datasets, though ENN has a slightly higher average ranking over the 20 datasets.

Figure 15 shows examples of varying k in all five algorithms. As expected, k has an influence on the predictive accuracy on every algorithm. In a specific application, the best k shall be searched in order to get the best accuracy.



(a) (b) (c) (d) (e) (f)

Fig. 15 Accuracies of algorithms with different k values. **a** Urban. **b** Ionosphere. **c** GPS. **d** Breast. **e** Heart. **f** Vote

6.2 Density-based clustering versus mass-based clustering

DBSCAN (Ester et al. 1996) is a natural choice for our evaluation not only because it is a commonly used clustering algorithm, but also it employs ϵ-neighbourhood density estimation (i.e., the key contender we used in the analysis in Sect. 4). Here we convert DBSCAN to MBSCAN, i.e., from density based to mass based, by simply replacing distance measure with mass-based dissimilarity, leaving the rest of the procedure unchanged. This effectively changes the use of ϵ-neighbourhood density estimation to μ-neighbourhood mass estimation, as described in Sect. 4.1. This enables a global threshold to be used in a mass distribution to identify all clusters of varying densities in a density distribution as shown in Figs. 11 and 12.

We evaluated MBSCAN and compared it with DBSCAN, SNN (Ertöz et al. 2003) and OPTICS (Ankerst et al. 1999). Note that the only difference among DBSCAN, SNN and MBSCAN is the dissimilarity matrix, which is preprocessed and serves as input to these algorithms. We used $iForest$ with the default setting (i.e., $\psi = 256$ and $t = 100$) (Liu et al. 2008) to generate the mass-based dissimilarity matrix as the input for MBSCAN.

For each clustering algorithm, the search range of either ϵ, μ or σ was from the minimum to the maximum value of pairwise dissimilarity in the given dataset. The search range of $MinPts$ in DBSCAN, SNN and MBSCAN was in the range $\{2, 3, \ldots, 10\}$. The parameter k in SNN was set to the square root of the data size as suggested by some researcher (Silverman 1986). For OPTICS, we searched $MinPts$ to produce the required hierarchical plots, and then searched threshold ξ[12] in the range $\{0.01, 0.02, \ldots, 0.99\}$ to extract clusters from each plot.

[12] Parameter ξ is used to identify downward and upward areas of a hierarchical plot in order to extract clusters. This hierarchical extraction method was proposed in the original OPTICS paper (Ankerst et al. 1999).

Table 10 Best F-measures of DBSCAN, OPTIC, SNN and MBSCAN on 20 datasets

Dataset	F-measure				Performance ratio		
	DBSCAN	OPTICS	SNN	MBSCAN	OPTICS	SNN	MBSCAN
Forest	0.26	0.29	**0.79**	0.57	1.08	3.00	2.16
ILPD	0.40	0.53	0.43	**0.54**	1.32	1.06	1.34
Messidor	0.48	0.49	0.48	**0.50**	1.03	0.99	1.03
Parkinson	0.36	0.33	**0.45**	0.44	0.93	1.24	1.21
Wilt	0.38	0.528	0.43	**0.53**	1.39	1.14	1.39
GPS	0.75	0.76	0.62	**0.81**	1.01	0.82	1.08
Phishing	0.40	0.43	**0.45**	0.40	1.07	1.12	1.01
QSAR	0.44	0.55	0.50	**0.61**	1.25	1.15	1.40
Spam	0.38	0.13	0.39	**0.47**	0.33	1.04	1.24
Urban	0.22	0.32	**0.46**	0.40	1.48	2.09	1.85
Libras	0.40	**0.52**	0.42	0.45	1.29	1.05	1.12
Thyroid	0.58	0.59	0.85	**0.86**	1.01	1.46	1.48
Segment	0.59	**0.70**	0.66	0.67	1.18	1.11	1.12
WDBC	0.60	0.68	0.70	**0.86**	1.13	1.17	1.44
Wine	0.65	0.76	**0.91**	0.90	1.18	1.40	1.39
Pendig	0.70	0.72	**0.82**	0.80	1.03	1.17	1.14
Seeds	0.75	0.80	**0.89**	0.889	1.07	1.19	1.19
Iris	0.87	0.85	0.96	**0.963**	0.98	1.10	1.11
S1	0.34	0.53	0.50	**0.62**	1.57	1.47	1.82
S2	0.94	0.98	0.99	**0.993**	1.05	1.06	1.06
#Top 1/geomean	0	2	7	11	1.12	1.29	1.33

The best performer on each dataset is boldfaced. The number of top 1 performances for each algorithm is given in the last row. Performance ratios of OPTICS, SNN, and MBSCAN with reference to DBSCAN on each of the 20 datasets are given in the last three columns; and the geometric mean (over 20 datasets) of the ratios for each algorithm is given in the last row

We recorded the best F-measure[13] of a clustering algorithm on a dataset. Because $iForest$ is a randomised method, we reported the average result over 10 trials.

Table 10 shows the best F-measure of DBSCAN, OPTICS, SNN and MBSCAN. It demonstrates that MBSCAN and SNN performed the best in 11 and 7 datasets, respectively. OPTICS only performed the best on 2 datasets. In terms of performance ratio with reference to DBSCAN, MBSCAN enhances DBSCAN the most by 33%; SNN enhances DBSCAN by 29%; and OPTICS enhances DBSCAN by 12%.

[13] Given a clustering result, we calculate the precision score Y_i and the recall score Z_i for each cluster based on the confusion matrix, and then the overall F-measure is the average over all clusters: F-measure$=\frac{1}{m}\sum_{i=1}^{m}\frac{2Y_iZ_i}{Y_i+Z_i}$. We investigated other clustering validation measures such as internal validation measure CVNN (Liu et al. 2013) and external validation measure AMI (Vinh et al. 2009) which have been shown to provide good measures for clustering algorithms that are not designed to identify noise, e.g., k-means. However, we found that these measures are not suitable for density-based clustering algorithms which can identify noise. This is because these formulations take into account assigned points only and ignore unassigned points. This can lead to a very good score when most points are unassigned. For example, SNN which obtains the best F-measure = 0.89 has 5% unassigned points on the Seeds dataset; whereas the best CVNN = 0.13 has 87% unassigned points.

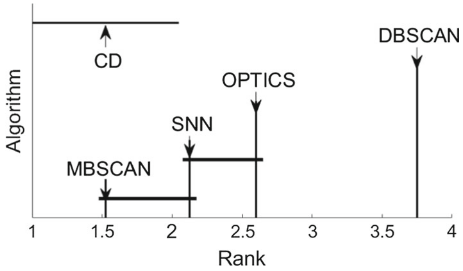


Fig. 16 Critical difference (CD) diagram of the post-hoc Nemenyi test ($\alpha = 0.05$) for the results showed in Table 10. The difference between two algorithms is significant if the gap between their ranks is larger than the CD. There is a line between two algorithms if the rank gap between them is smaller than the CD

Table 11 Time and space complexities of dissimilarity matrix calculation based on ℓ_p, m_e and SNN

Measure	Time complexity	Space complexity
ℓ_p	$O(dn^2)$	$O(dn)$
m_e	$O(t\psi log\psi + n^2 tlog\psi)$	$O(t\psi log\psi + dn)$
SNN	$O(k^2n^2 + dn^2)$	$O(dn + kn)$

Although MBSCAN and SNN have similar F-measures in many datasets, one is better than the other on a few datasets. MBSCAN is significantly better than SNN in two datasets: S1 and WDBC. For example, MBSCAN enhances DBSCAN by more than 80% compared with less than 50% achieved by SNN on S1. The reverse is true on the Forest dataset, where SNN enhances DBSCAN by 300% compared with 216% by MBSCAN. MBSCAN and SNN enhance DBSCAN by more than 15% on 12 and 10 datasets, respectively. OPTICS only does so on 8 datasets.

The post-hoc Nemenyi test, shown in Fig. 16, demonstrates that MBSCAN is the only algorithm which performs significantly better than OPTICS and DBSCAN. SNN is significantly better than DBSCAN, but not better than OPTICS.

6.3 Evaluation in runtime

For unsupervised learning tasks, the only difference between the ℓ_p and m_e versions of the algorithms is the computation time for the dissimilarity matrix. After the matrix is computed and served as input, the algorithm has the same runtime regardless of the dissimilarity used to compute the matrix.

Using ℓ_p to compute the dissimilarity matrix has $O(dn^2)$ time complexity. m_e builds $iForest$ and computes the dissimilarity matrix based on $iForest$, which yields $O(t\psi log\psi + n^2 tlog\psi)$ time complexity. For large datasets, $\psi \ll n$, the time cost is $O(n^2)$. The time complexity of SNN similarity is $O(n^3)$ in the worst case when $k = \sqrt{n}$ or larger. Table 11 gives the time and space complexities of dissimilarity matrix calculation based on ℓ_p, m_e and SNN.

Table 12 shows the runtime of the dissimilarity matrix calculation for the three dissimilarities on three datasets. In small dimensional datasets, m_e has almost the same runtime as SNN, but takes longer to compute than ℓ_p due to the use of $iForest$. However, in large and high dimensional datasets such as p53Mutant, m_e is much faster than both ℓ_p and SNN because it is independent of the data dimensionality, i.e., each split node of a tree chooses one attribute randomly up to the certain height limit only.

For supervised learning tasks, the current implementation of m_e using $iForest$ demands a traversal of all the trees for every measurement of dissimilarity between a test point and one point in the training set. This is more costly than the ℓ_p measurement. For example, on the

Table 12 Runtime of the dissimilarity matrix calculation (in seconds)

Data set (Data size) (Dimension)	Segment (2310) (19)	Pendig (10992) (16)	p53Mutant (10387) (5408)
ℓ_p	5	110	8182
m_e	31	600	548
SNN	26	573	9141

Corel dataset used in Sect. 6.1, kNN took 1700 s while kLMN took 116,600 s to complete the 5-fold cross-validation experiment.

Thus, a more efficient implementation of m_e is required for supervised learning tasks.

7 Related data dependent dissimilarities and metric axioms

7.1 m_p and $\ell_{p,cdf}$

The first version of mass-based dissimilarity is called m_p (Aryal et al. 2014a) and it is defined as follows:

$$m_p(\mathbf{x}, \mathbf{y}) = \left(\frac{1}{q} \sum_{i=1}^{q} \left(\frac{|R_i(\mathbf{x}, \mathbf{y})|}{n} \right)^p \right)^{\frac{1}{p}} \tag{9}$$

where $|R_i(\mathbf{x}, \mathbf{y})| = |\{z_i : \min(x_i, y_i) - \delta \leq z_i \leq \max(x_i, y_i) + \delta\}|$ is the data mass in dimension-i region $R_i(\mathbf{x}, \mathbf{y}) = [\min(x_i, y_i) - \delta, \max(x_i, y_i) + \delta]$, $\delta \geq 0$; $p > 0$; q is the number of dimensions; and n is the number of points in the given dataset.

Note that this formulation is the same as ℓ_p, except that the data mass in $R_i(\mathbf{x}, \mathbf{y})$ is used instead of distance between $\mathbf{x}$ and $\mathbf{y}$ in each dimension. This formulation is a special case of m_e which considers data mass in individual dimensions independently.

The single dimensional implementation of m_p has the advantage that it can be easily extended to deal with categorical attributes and mixed attribute types. There is no obvious way to deal with these attributes in the $iForest$ implementation. The disadvantage is that m_p can perform poorly on datasets in which there is strong dependency between multiple attributes. This is where the $iForest$ implementation of m_e can perform better than m_p.

Transformation using cdf One can view m_p (Aryal et al. 2014a) as almost equivalent to ℓ_p applied to a transformed data set D', where $x'_i = cdf(x_i)$, i.e., points are represented by their cumulative distribution function in each dimension. Note that

$$m_p(x_i, y_i) = P(x_i \leq X_i \leq y_i) = cdf(y_i) - cdf(x_i) + P(x_i)$$

whereas $\ell_p(x'_i, y'_i) = P(x_i < X_i \leq y_i) = cdf(y_i) - cdf(x_i)$.

If every point is unique, then the difference between m_p and ℓ_p in dimension i is $P(x_i) = 1/n$, or the self-dissimilarity is constant. However, the difference between m_p and ℓ_p enlarges when there are duplicate points.[14] Often in real high-dimensional datasets, many points can have the same value in many dimensions, e.g., many documents in a collection have the same occurrence frequency of a term; or different individuals have the same age. In the extreme case where all the points have the same value in dimension i, $m_p(x_i, x_i) = \frac{n}{n} = 1$ (maximal dissimilarity) whereas $\ell_p(x'_i, x'_i) = cdf(x_i) - cdf(x_i) = 0$ (minimal dissimilarity).

[14] When a histogram is used to estimate cdf, it has the same effect of creating duplicates in each bin.

Table 13 Compliance with the metric axioms. Note that for $\ell_{p,cdf}$ or ℓ_p to be a metric, $p \geq 1$

	m_e	m_p	$\ell_{p,cdf}$ or ℓ_p
Constancy of self-dissimilarity	×	×	✓
Minimality of self-dissimilarity	×	×	✓
Symmetry	✓	✓	✓
Triangular inequality	✓	✓	✓

Table 14 Different types of dissimilarities

	Data dependent	Data independent
Metric	$\ell_{p,cdf}$; Mahalanobis; Lin's measure	ℓ_p $(p \geq 1)$
Pseudo-metric	Distance metric learning	
Semi-metric		ℓ_p $(0 < p < 1)$
Datad-metric	m_e; m_p	

Note that ℓ_p with the cdf transformation ($\ell_{p,cdf}$) is a measure in-between data independent ℓ_p and data dependent m_p, i.e., it becomes data dependent for $\forall$ $\mathbf{x} \neq \mathbf{y}$ only and its self-dissimilarity is zero and constant. As a result, it is still a metric.

7.2 Compliance with metric axioms

A comparison of m_e and m_p in terms of compliance with the metric axioms is given in Table 13. m_e is in compliance with the triangular inequality and symmetry axioms when the $iForest$ implementation is used.

Let a, b, c be values of a real attribute, and $a < b < c$. A binary $iTree$ will partition the points as follows: either $R(a, b) \subseteq R(a, c)$ or $R(b, c) \subseteq R(a, c)$. Because either $R(a, b) = R(a, c)$ or $R(b, c) = R(a, c)$, it results in $|R(a, b)| + |R(b, c)| > |R(a, c)|$, satisfying the triangular inequality condition.

It is interesting to note that the smallest region in mass-based dissimilarity is analogous to the shortest path in distance metric; and both of them lead to triangular inequality.

m_e is symmetric because $R(a, b)$ and $R(b, a)$ occupy the same node in a tree.

m_e is not in compliance with the first two axioms because of the properties described in Table 1. Because of its unique feature of data dependent self-dissimilarity, we name it *datad-metric* to differentiate it from existing measures such as quasi-metric, meta-metric, semi-metric, peri-metric (which violate one or two of the metric axioms).

As our focus is on data dependent measures, the comparison with generalised data independent measures (e.g., ℓ_p, where $0 < p < 1$, that violates the triangular inequality axiom) is outside the scope of this paper. Interested readers of generalised data independent measures may refer to Jacobs et al. (2000), Tan et al. (2009). Whether these generalised data independent measures can overcome the shortcomings of the four algorithms we have studied here (including those shown in the Appendices) is an interesting research topic in the future.

There are other existing data dependent dissimilarities such as Mahalanobis distance (Mahalanobis 1936) and Lin's probabilistic measure (Lin 1998). As they are metrics, their self-dissimilarity is constant, unlike m_e.

A summary of different types of dissimilarities is provided in Table 14. We describe the relation to distance metric learning, data dependent kernel and similarity-based learning in the next section.

8 Relation to distance metric learning, data dependent kernel and similarity-based learning

8.1 Distance metric learning

Distance metric learning can be viewed as a restricted form of data dependent dissimilarity. Typically, the aim is to learn a generalised (or parameterised) Mahalanobis distance, subject to some optimality constraint, from a dataset. In simple terms, the aim is to reduce the dissimilarity between points of the same class; and increase the dissimilarity between points of different classes.

Wang and Sun (2015) define a common form of distance metric learning as:

"The problem of learning a distance function ∂ for a pair of data points $\mathbf{x}$ and $\mathbf{y}$ is to learn a mapping function f, such that $f(\mathbf{x})$ and $f(\mathbf{y})$ will be in the Euclidean space and $\partial(\mathbf{x}, \mathbf{y}) = ||f(\mathbf{x}) - f(\mathbf{y})||$, where $|| \cdot ||$ is the ℓ_2 norm."

A more general formulation must still conform to some norm and is a pseudo-metric, i.e., it relaxes one metric condition from "$\partial(\mathbf{x}, \mathbf{y}) = 0$ iff $\mathbf{x} = \mathbf{y}$" to "if $\mathbf{x} = \mathbf{y}$, then $\partial(\mathbf{x}, \mathbf{y}) = 0$", in additional to the learning requirement which needs to know the class label for each point.

In contrast, mass-based dissimilarity, without learning, derives the dissimilarity of $\mathbf{x}$ and $\mathbf{y}$ directly from data based on probability mass of the region covering $\mathbf{x}$ and $\mathbf{y}$, without class information.

Mass-based dissimilarity is conceptually simpler, yet more generic, than distance metric learning; and it is a data dependent measure that is not restricted to a single form (which requires Euclidean distance, for example).

The implementations of mass-based dissimilarity differ in how the region is defined, including one similar to ℓ_p-norm called m_p dissimilarity (Aryal et al. 2014a), which permits all valid p values including $p = 2$.

In other words, distance metric learning exploits information from Euclidean metric space, with a minor relaxation to become a pseudo metric. The focus is to learn a mapping from data based on some optimality criterion. It is an indirect way to get a restricted form of data-dependent dissimilarity. In contrast, mass-based dissimilarity has no such insistence or assumption, and it derives the dissimilarity directly from data, resulting in a generic form of data-dependent dissimilarity.

The empirical comparison between m_e and two distance metric learning methods [i.e., large margin nearest neighbour (Weinberger and Saul 2009) and GMML (Zadeh et al. 2016)] has been shown in Sect. 6.1. The former is unsupervised which requires neither labelled data nor learning; the latter is supervised which requires labelled data and learning. Even with the advantage of class information and learning, distance metric learning is not found to be significantly better than m_e.

A summary of the key differences between mass-based dissimilarity and distance metric learning is provided in Table 15. In a nutshell, less is more in solving the deficiency of distance measures, i.e., one gets more out of mass-based dissimilarity than distance metric learning with significantly less computational requirements.

8.2 Data dependent kernel

Designing a good kernel function is at the heart of kernel methods. The use of a poorly designed kernel function severely affects a kernel method's predictive accuracy. One of the first methods to adapt a kernel function to the structure of the data is through a conformal

Table 15 Key differences between mass-based dissimilarity and distance metric learning

	Mass-based dissimilarity	Distance metric learning
Conceptual basis	Probability mass of the region covering $\mathbf{x}$ and $\mathbf{y}$	Learn mapping f such that $\partial(\mathbf{x}, \mathbf{y}) = \|\|f(\mathbf{x}) - f(\mathbf{y})\|\|_2$ in order to reduce the intra-class distance and increase the inter-class distance
Foundation	Mass estimation	Generalised Mahalanobis distance
Metric	Datad-metric	Pseudo-metric
Learning	No learning required: the mass estimation is based on the smallest region covering $\mathbf{x}$ and $\mathbf{y}$. No class information is required	Need learning: optimising some criterion based on either the entire dataset or a local region of the data. Need class information
Connection	Connection to cdf transform in one formulation (m_p). See Sect. 7.1	Strong connection to dimensionality reduction: Euclidean distance in some embedding space

transformation of kernel functions (Amari and Wu 1999; Wu and Amari 2002; Xiong et al. 2007). In the classification context, the idea is to modify the kernel function such that the spatial distances around the boundary between two classes are enlarged (and those outside the boundary region reduced). To achieve this objective, some knowledge of the boundary is required in order to learn a data dependent kernel for a given dataset. This objective bears some resemblance to that of distance metric learning (described in the last subsection), i.e., to reduce the distance between points of the same class and increase the distance between points of different classes. Both require a computationally expensive optimisation learning process. The resultant dissimilarity measure of a data dependent kernel is metric. Here, the data dependency relies on class information.

In contrast, mass-based dissimilarity is derived directly from the given dataset, requiring neither learning nor information about class labels. The term ‘data dependent’ is used for mass-based dissimilarity in a more general context (than data dependent kernel or distance metric learning) to denote the dependency on data distribution without the class information.

A point worth noting is that a kernel family needs to be determined by a user of kernel methods, even if a kernel can be made data dependent. The equivalent requirement for mass-based dissimilarity is the model (or the partitioning strategy) used to define regions. It is arguably easier to determine the latter than the former.

The early development of data dependent kernel can be traced back to learning an optimal global nearest neighbour metric (Short and Fukunaga 1981; Fukunaga and Flick 1984). Fukunaga (1990) (pages 318–319) has commented that it is unclear how to estimate the metric such that the bias could be minimised because of the requirement of a positive definite matrix (a similar requirement of many kernel methods).

8.3 Similarity-based learning

The key concern of similarity-based (non-metric) learning (Chen et al. 2009; Schleif and Tino 2015) is: how to deal with the given, often naturally obtained, dissimilarity which is non-metric (e.g., violations of symmetry and/or triangular equality); and the data are often not represented in a vectorial representation. In this setting, two general approaches (Schleif and Tino 2015) are: (i) Transforming non-metric dissimilarity to metric dissimilarity so that existing methods which rely on the constraint of metric can be applied; and (ii) creating learning methods that deal with the given non-metric dissimilarity directly. Most of these methods are data independent, i.e., the dissimilarity between two points depends on these two points only.

The current framework of mass-based dissimilarity (and also distance metric learning and data dependent kernel described in the last two subsections) assumes that points are given in a vectorial representation. With this assumption, there is a choice in using either metric or non-metric dissimilarity; albeit the default choice is often metric by using distance measures (as in the cases of distance metric learning and data dependent kernel). The mass-based dissimilarity offers an unconventional choice to employ data dependent non-metric dissimilarity under this assumption.

In other words, the premises of mass-based dissimilarity and similarity-based learning are different. A comparison with similarity-based learning is only meaningful when mass-based dissimilarity is further developed into one which converts the given (data independent) non-metric dissimilarity into a data dependent dissimilarity. Whether this conversion necessitates a learning process is an open question.

Having said that, there is still one distinguishing feature of mass-based dissimilarity: it is a general approach which can be used for a variety of applications. In contrast, many similarity-based learning methods are based on application-specific studies, where the naturally obtained dissimilarity is application specific.

9 Discussion

9.1 Concepts

Except in the case of given non-metric dissimilarity, dissimilarity measures are assumed to be a metric or a pseudo-metric as a necessary criterion for all data mining tasks. This work shows that requiring the metric assumptions may be an impediment to producing good performing models in two tasks. The fact that mass-based dissimilarity m_e, which violates two metric axioms, can be used to overcome the shortcomings of two algorithms demonstrates the inadequacy of metric axioms. The data dependent property is the overarching factor which leads to this outcome.

We imply that distance measures are the root cause of key shortcomings in two existing algorithms, highlighted in Table 5. Having identified the root cause and created an effective alternative to distance measure, the solution becomes simple—merely replacing the distance measure with the mass-based dissimilarity; the otherwise unchanged algorithm can now overcome its shortcoming.

The result of not recognising the root cause can often lead to a solution which is more complicated than necessary and may not resolve the issue completely. An example is the inability of density-based clustering to find all clusters of varying densities. This issue is well-known and many suggestions have focused on density-based solutions (Ankerst et al. 1999; Ertöz et al. 2003). The fact that the ϵ-neighbourhood density estimator employed relies on distance measure, which is the root cause of the shortcoming, has been overlooked.

It is interesting to note that one of the existing solutions, i.e., SNN clustering has been incorrectly designated as density-based (Ertöz et al. 2003; Tan et al. 2005) thus far. Our analysis in Sect. 4.4 has unveiled that when replacing SNN (dis)similarity with the distance measure in ϵ-neighbourhood density estimator, the result is a mass estimator, not a density estimator.

μ-neighbourhood mass can be viewed as a general version of ϵ-neighbourhood density. The shapes and volumes of ϵ-neighbourhood density regions are fixed for a given ϵ. In contrast, the shapes and volumes of μ-neighbourhood mass regions depend on data distribution. We provide an example in Fig. 8 that μ-neighbourhood mass has a regular shape and fixed volume region like ϵ-neighbourhood estimator in uniform density distribution only.

9.2 Implementations

The use of $iForest$ can be viewed as estimating probability from multiple variable-size multi-dimensional histograms.

Parameter ψ in $iForest$, used in the μ-neighbourhood estimator, is a smoothing parameter similar to k in a k-nearest neighbour density estimator. High ψ yields large trees which are sensitive to local variations in data distribution—similar effect of setting small k.

Since the default setting of $iForest$ ($\psi = 256$ and $t = 100$) can be used to provide good performance on many datasets, the implementation of mass-based dissimilarity based on $iForest$ does not create additional limitations or parameters that need to be tuned.

The generic formulation of mass-based dissimilarity allows different implementations, including different variants of $iForest$ (see Sect. 2.4); and m_p dissimilarity (Aryal et al. 2014a) and SNN (dis)similarity are its special cases—all of them possess the characteristic of judged dissimilarity as prescribed by psychologists (Krumhansl 1978).

It is possible to use SNN in kNN algorithms (e.g., kNN anomaly detection, and kNN and MLkNN classifications). However, its use has two issues. First, there are two k parameters as kNN is employed separately in the dissimilarity calculation and the decision making process. Second, the high time complexity shown in Table 11 makes SNN prohibitive in large datasets.

Note that a mass-based neighbourhood function can be implemented using a distance measure, as in the case of SNN. But, it is not only an indirect way to estimate mass but also an expensive one, as mentioned in Sect. 4.4.

9.3 Limitations of current implementations

The $iForest$ implementation of mass-based dissimilarity has five limitations. First, the time complexity is higher than distance measure in supervised learning, as described in Sect. 6.3. Second, it is limited to numeric attributes only. Third, like all tree implementations, it can deal with low- and medium-size dimensionality only because each tree considers only a small subset of available attributes. Fourth, the current implementations (either m_e or m_p) produce datad-metric only. These do not suit applications which demand violations of triangle inequality and/or symmetry. Fifth, the current implementations assume vectorial representation of data points. As such, it is unclear how mass-based dissimilarity can be created for applications which have no vectorial representations and only dissimilarities between points are given.

The SNN implementation is not a good alternative to $iForest$ because of its high time complexity; and it introduces an additional parameter k which is sensitive and needs to be tuned. This is despite the fact that it may be able to better deal with categorical attributes and high-dimensional datasets.

The m_p implementation simplifies to single-dimensional probability estimations. Its current implementation [shown in Eq. (9)] requires a range search in each dimension which costs

$O(d \log n)$ by using a binary search tree for each dimension (compared against $O(d)$ for ℓ_p.) The time complexity has the potential to be further reduced. In addition, this implementation is also more readily applied to categorical attributes and high dimensional datasets. However, the key shortcoming is that every dimension is being considered independently in probability estimation. This can have a negative impact in many real-world applications where the dependency between attributes needs to be considered.

9.4 Probabilistic kNN classifiers and related techniques

Various techniques have been developed over the years to tackle some of the limitations of the kNN classifier, such as its sensitivity to high dimensions or to settings of the parameter k.

For instance, the probabilistic kNN (PNN) model (Holmes and Adams 2002) made use of Bayesian techniques to remove the need to choose k. The PNN model can be seen as using hold-one-out cross-validation to generate a smoothed ensemble of kNN classifiers over different values for k (weighted by their posterior probabilities).

The Discriminant Adaptive Nearest Neighbor (DANN) model (Hastie and Tibshirani 1996) attempts to deal with the susceptibility of kNN classifiers to high-dimensionality by making use of ideas from Linear Discriminant Analysis. More specifically, the model modifies the distance metric at points near class boundaries such that the k-nearest neighbourhood is warped and stretches out along the boundaries. The ellipsoid-shaped neighborhoods mean that points on the other side of the boundary are less likely to fall within the neighborhood and thus allow for more reliable determination of the true class boundary in high dimensions. We note that the method does not, to the best of our knowledge, deal with the issue of differences in density on either side of the class boundary.

Finally, the Bayesian Adaptive Nearest Neighbor (BANN) model (Guo and Chakraborty 2010) combines the ideas of both PNN and DANN into a single model.

Overall, these techniques can be seen as using ensembling method to remove the dependence on k and distance-warping methods to reduce bias at points close to the decision boundary. We note that modified distance measure, while data dependent, makes use of the class label information, thus distinguishing the technique from mass-based approaches outlined in this paper. Furthermore, the computational requirements of the techniques are substantial due to the need to perform both leave-one-out cross-validation and matrix inversion.

10 Concluding remarks

We introduce a generic mass-based dissimilarity which is readily applied to existing algorithms in different tasks. The mass-based dissimilarity implemented with $iForest$ overcomes key shortcomings of two existing algorithms that rely on distance, and effectively improves their task-specific performance on distance-based classification and density-based clustering.

These existing algorithms are transformed by simply replacing the distance measure with the mass-based dissimilarity, leaving the rest of the procedures unchanged.

As the transformation heralds a fundamental change of perspective in finding the closest match neighbourhood, the converted algorithms are more aptly called lowest probability mass neighbour algorithms than nearest neighbour algorithms, since the lowest mass represents the most similar.

Our analyses provide an insight into the conditions under which the distance-based neighbourhood methods fail and the mass-based neighbourhood methods succeed in classification and clustering tasks.

The proposed mass-based dissimilarity has a unique feature in comparison with existing data dependent measures, that is, its self-dissimilarity is data dependent and not constant. We call it datad-metric, as opposed to existing data dependent metrics and generalised data independent metrics (e.g., quasi-metric, meta-metric, semi-metric, peri-metric).

We disclose that two of the four metric axioms are not necessary in developing a model that performs well. This opens up research to (i) other forms of data dependent dissimilarities that work in practice, and (ii) incorporating learning with datad-metrics, where the research effort thus far has been largely restricted to metric learning and non-metric learning; and the latter is largely data independent. In addition, we will explore other implementations of mass-based dissimilarity and investigate their influence in different data mining tasks.

Acknowledgements This material is based upon work partially supported by the Air Force Office of Scientific Research, Asian Office of Aerospace Research and Development (AOARD) under Award Number: #FA2386-17-1-4034 and 111 Project (B14020) (Kai Ming Ting); and NSFC (61751306, 61333014) and the Collaborative Innovation Center of Novel Software Technology and Industrialization (Zhi-Hua Zhou). The preliminary version of this paper was published in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Ting et al. 2016).

Appendix

A Algorithm to generate random trees - *iForest*

$iForest$ (Liu et al. 2008) consists of t $iTrees$, each built independently using a subset $\mathcal{D}$, sampled without replacement from D, where $|\mathcal{D}| = \psi$. The maximum tree height $h = \lceil log_2\psi \rceil$. Note that the parameter e in $iTree$ is initialised to 0 at the beginning of the tree building process.

```
Algorithm 1 $iTree(X, e, h)$
Require: $X$ - input data; $e$ - current height; $h$ - height limit.
Ensure: an $iTree$.
if $e \geqslant h$ OR $|X| \leqslant 1$ then
    return $exNode\{Size \leftarrow |X|\}$;
else
    Randomly select an attribute $q$;
    Randomly select a split point $p$ between $min$ and $max$ values of attribute $q$ in $X$;
    $X_l \leftarrow filter(X, q < p), X_r \leftarrow filter(X, q \geqslant p)$;
    return $inNode\{ Left \leftarrow iTree(X_l, e+1, h),$
                     $Right \leftarrow iTree(X_r, e+1, h),$
                     $SplitAttr \leftarrow q, SplitValue \leftarrow p\}$;
end if
```

B Proof for Theorem 1

Proof Let $v_T(\mathbf{x})$ and $v_S(\mathbf{x})$ denote the volumes of the intersections $B(\mathbf{x}) \cap \mathcal{X}_T$ and $B(\mathbf{x}) \cap \mathcal{X}_S$, respectively. Then, $\rho_T(\mathbf{x})v_T(\mathbf{x}) + \rho_S(\mathbf{x})v_S(\mathbf{x}) \simeq k/N$ holds, and $k_T(\mathbf{x})$ follows a binomial distribution $\mathrm{B}(k, pr)$ where $pr = \rho_T(\mathbf{x})v_T(\mathbf{x})/\ (\rho_T(\mathbf{x})v_T(\mathbf{x}) + \rho_S(\mathbf{x})v_S(\mathbf{x}))$ which is the probability that a positive instance appears in $NN_k(\mathbf{x})$. Recall that $\rho_T(\mathbf{x})/\rho_S(\mathbf{x}) > 1$ holds from the assumption: $\forall \mathbf{x} \in \mathcal{X}_T, \forall \mathbf{y} \in \mathcal{X}_S, P(\mathbf{x}) > P(\mathbf{y})$. The proof is provided in two parts: (a) the misclassification rate in $\mathcal{X}_S$: ε_S; and (b) the misclassification rate in $\mathcal{X}_T$: ε_T.

(a) For any border point $\mathbf{x} \in D_S$, $v_S(\mathbf{x}) > v_T(\mathbf{x})$ holds since the shape of $B(\mathbf{x})$ is symmetric in the space and the border between $\mathcal{X}_T$ and $\mathcal{X}_S$ is almost straight in $B(\mathbf{x})$ by the assumption. Thus, there exists $\alpha_S(\mathbf{x}) > 1$ for any border point $\mathbf{x} \in D_S$ such that $v_S(\mathbf{x}) = \alpha_S(\mathbf{x})v_T(\mathbf{x})$. This implies that $\rho_S(\mathbf{x})v_S(\mathbf{x}) < \rho_T(\mathbf{x})v_T(\mathbf{x})$, *i.e.*, $pr = \rho_T(\mathbf{x})v_T(\mathbf{x})/(\rho_T(\mathbf{x})v_T(\mathbf{x}) + \rho_S(\mathbf{x})v_S(\mathbf{x})) > 0.5$ in $\mathrm{B}(k, pr)$ of $k_T(\mathbf{x})$, holds for any border point $\mathbf{x} \in D_S$, if $1 < \alpha_S(\mathbf{x}) < \rho_T(\mathbf{x})/\rho_S(\mathbf{x})$.

Given a border point $\mathbf{x} \in D_S$ where $\alpha_S(\mathbf{x})$ satisfies the above inequality, $P(k_T(\mathbf{x}) > k/2) > 0.5$ holds because of $pr > 0.5$ in $\mathrm{B}(k, pr)$ of $k_T(\mathbf{x})$, and thus such a border point $\mathbf{x}$ is most probably misclassified.

For such $\mathbf{x}$, if $\rho_T(\mathbf{x})$ increases while $\rho_S(\mathbf{x})$ is constant, the radius of $B(\mathbf{x})$ decreases or remains constant, *i.e.*, $v_S(\mathbf{x})$ decreases or remains constant. Hence, $\rho_T(\mathbf{x})v_T(\mathbf{x})$ increases or remains constant because $\rho_T(\mathbf{x})v_T(\mathbf{x}) \simeq k/N - \rho_S(\mathbf{x})v_S(\mathbf{x})$. Similarly, if $\rho_S(\mathbf{x})$ decreases while $\rho_T(\mathbf{x})$ is constant, the radius of $B(\mathbf{x})$ increases or remains constant, *i.e.*, $v_T(\mathbf{x})$ increases or remains constant. Hence, $\rho_S(\mathbf{x})v_S(\mathbf{x})$ decreases or remains constant because of $\rho_S(\mathbf{x})v_S(\mathbf{x}) \simeq k/N - \rho_T(\mathbf{x})v_T(\mathbf{x})$.

Accordingly, $pr = \rho_T(\mathbf{x})v_T(\mathbf{x})/(\rho_T(\mathbf{x})v_T(\mathbf{x}) + \rho_S(\mathbf{x})v_S(\mathbf{x}))$ becomes higher as $\rho_T(\mathbf{x})/\rho_S(\mathbf{x})$ increases; and the probability $P(k_T(\mathbf{x}) > k/2)$ becomes higher too. In other words, such a border point $\mathbf{x} \in D_S$ is probabilistically misclassified as belonging to T because $NN_k(\mathbf{x})$ most probably has $k_S(\mathbf{x}) < k_T(\mathbf{x})$.

In addition, if the upper bound of $\alpha_S(\mathbf{x})$, *i.e.*, $\rho_T(\mathbf{x})/\rho_S(\mathbf{x})$, increases, then more points $\mathbf{x} \in D_S$ have chances to be misclassified as belonging to T. This is because the larger $\alpha_S(\mathbf{x})$ is, the larger $v_S(\mathbf{x})$ and smaller $v_T(\mathbf{x})$ are in $B(\mathbf{x})$. This means that border points $\mathbf{x} \in D_S$ which are farther from the border between $\mathcal{X}_T$ and $\mathcal{X}_S$ are most probably misclassified, as $\rho_T(\mathbf{x})/\rho_S(\mathbf{x})$ increases.

Since the above argument holds for any border point $\mathbf{x} \in D_S$, the kNN classifier's misclassification rate in $\mathcal{X}_S$: ε_S is a probabilistically increasing function of the density ratio $\bar{\rho}_T/\bar{\rho}_S$.

(b) Similarly, there exists $\alpha_T(\mathbf{x}) > 1$ for any border point $\mathbf{x} \in D_T$ such that $v_T(\mathbf{x}) = \alpha_T(\mathbf{x})v_S(\mathbf{x})$. This implies that $\rho_S(\mathbf{x})v_S(\mathbf{x}) > \rho_T(\mathbf{x})v_T(\mathbf{x})$, *i.e.*, $pr < 0.5$ in $\mathrm{B}(k, pr)$ of $k_T(\mathbf{x})$, holds if $1 < \alpha_T(\mathbf{x}) < \rho_S(\mathbf{x})/\rho_T(\mathbf{x})$. If $\alpha_T(\mathbf{x})$ of a point $\mathbf{x} \in D_T$ is in this interval, such $\mathbf{x}$ is most probably misclassified as belonging to $\mathcal{X}_S$. However, such $\alpha_T(\mathbf{x})$ does not exist, because its upper bound $\rho_S(\mathbf{x})/\rho_T(\mathbf{x})$ is less than 1 by the assumption $\rho_S(\mathbf{x}) < \rho_T(\mathbf{x})$. Thus, almost no points in $\mathcal{X}_T$ have chances to be misclassified, and the kNN classifier's misclassification rate in $\mathcal{X}_T$: ε_T is always most probably zero. □

C Proof for Theorem 2

Let $\mathbf{z}_e$, $\mathbf{z}_n$ and $\mathbf{z}$ be sets of effective and non-effective dimensions of $R(\mathbf{x}, \mathbf{y}|H; D)$ and their union, respectively. Then, under a given set of hierarchical partitions $\mathcal{H}(D)$ of the dataset D, $m_e(\mathbf{x}, \mathbf{y})$ is represented by an effective mass-based dissimilarity $\tilde{m}_e(\mathbf{x}, \mathbf{y})$ with an $\ell_p(\mathbf{x}, \mathbf{y})$ component as follows.

$$\begin{aligned} m_e(\mathbf{x}, \mathbf{y}) &= E_{\mathcal{H}(D)}\left[\int_{R(\mathbf{x},\mathbf{y}|H;D)} \rho(\mathbf{z})\mathrm{d}\mathbf{z}_e \mathrm{d}\mathbf{z}_n\right] \\ &= E_{\mathcal{H}(D)}\left[\int_{R(\mathbf{x},\mathbf{y}|H;D)} \rho(\mathbf{z})\mathrm{d}\mathbf{z}_e\right] \prod_{z\in\mathbf{z}_n} E_{\mathcal{H}(D)}[L(z)] \\ &= \tilde{m}_e(\mathbf{x}, \mathbf{y})\tilde{C}_R \ell_p(\mathbf{x}, \mathbf{y})^{q-\tilde{q}}, \end{aligned}$$

where $E_{\mathcal{H}(D)}[\cdot]$ represents a statistical expectation taken over $\mathcal{H}(D)$, $L(z)$ is the size of dimension z of $R(\mathbf{x}, \mathbf{y}|H; D)$, $\tilde{m}_e(\mathbf{x}, \mathbf{y}) = E_{\mathcal{H}(D)}[\int_{R(\mathbf{x},\mathbf{y}|H,D)} \rho(\mathbf{z})\mathrm{d}\mathbf{z}_e]$ does not depend on $\mathbf{z}_n$, and $\tilde{C}_R$ is a constant. The second line holds, because $\rho(\mathbf{x})$ does not depend on the non-effective dimensions $\mathbf{z}_n$, and each dimension of $R(\mathbf{x}, \mathbf{y}|H; D)$ is independently defined from the others. The third line holds, since $E_{\mathcal{H}(D)}[L(z)]$ of every dimension is proportional to $\ell_p(\mathbf{x}, \mathbf{y})$ defining the size of $R(\mathbf{x}, \mathbf{y}|H; D)$. This yields the following expression of $m_e(\mathbf{x})$.

$$m_e(\mathbf{x}) = \max_{\mathbf{y}\in LMN_k(\mathbf{x})} m_e(\mathbf{x}, \mathbf{y}) = \tilde{m}_e(\mathbf{x})\tilde{C}_R \ell_p(\mathbf{x})^{q-\tilde{q}}, \tag{10}$$

where $\tilde{m}_e(\mathbf{x})$ and $\ell_p(\mathbf{x})$ are $\tilde{m}_e(\mathbf{x}, \mathbf{y})$ and $\ell_p(\mathbf{x}, \mathbf{y})$ for $\mathbf{y}$ yielding the maximum $m_e(\mathbf{x}, \mathbf{y})$, respectively.

Then, we further introduce the following two definitions.

$$\tilde{E}_{\mathcal{H}(D)}[\bar{\rho}(R(\mathbf{x}, \mathbf{y}|H; D))] = \frac{E_{\mathcal{H}(D)}[\int_{R(\mathbf{x},\mathbf{y}|H;D)} \rho(\mathbf{z})\mathrm{d}\mathbf{z}]}{E_{\mathcal{H}(D)}[\int_{R(\mathbf{x},\mathbf{y}|H;D)} \mathrm{d}\mathbf{z}]}, \text{ and} \tag{11}$$

$$\bar{\rho}(V(\mathbf{x}, \mathbf{y})) = \frac{\int_{V(\mathbf{x},\mathbf{y})} \rho(\mathbf{z})\mathrm{d}\mathbf{z}}{\int_{V(\mathbf{x},\mathbf{y})} \mathrm{d}\mathbf{z}}. \tag{12}$$

Based on these definitions, we derive the following two lemmas.

Lemma 1 *Given a hierarchical partitioning $H \in \mathcal{H}(D)$ of a dataset D and a border point $\mathbf{x}$ located in $\mathcal{X}_A$ under $LMN_k(\mathbf{x})$ provided by H, let the probability mass in $R^i_A(\mathbf{x})$, $R^o_A(\mathbf{x})$ and $R_B(\mathbf{X})$ be $P(R^i_A(\mathbf{x}))$, $P(R^o_A(\mathbf{x}))$ and $P(R_B(\mathbf{X}))$, respectively. Then, the following upper-bound $\phi(\mathbf{x})$ exists:*

$$\frac{P(R^o_A(\mathbf{x})) + P(R_B(\mathbf{X}))}{P(R^i_A(\mathbf{x}))} \leq \phi(\mathbf{x})$$

where

$$\phi(\mathbf{x}) = \sup_{\mathbf{y}\in G(\mathbf{x})} \frac{\bar{\rho}(V(\mathbf{x}, \mathbf{y}))}{\rho_A(\mathbf{x})} \left\{\frac{\rho_A(\mathbf{x})}{\tilde{E}_{\mathcal{H}}[\bar{\rho}(R(\mathbf{x}, \mathbf{y}|H, D))]}\right\}^{1/\tilde{q}}$$

and $G(\mathbf{x})$ is the edge of $R(\mathbf{x})$ belonging to $R^o_A(\mathbf{x}) \cup R_B(\mathbf{X})$.

Proof Proof. From Eq. (11), because of the mutual independence of the dimensions of $R(\mathbf{x}, \mathbf{y}|H; D))$ and $\ell_p(\mathbf{x})$ defining the size of $R(\mathbf{x}, \mathbf{y}|H; D))$, we obtain the following expression:

$$\begin{aligned} m_e(\mathbf{x}) &= E_{\mathcal{H}(D)}\left[\int_{R(\mathbf{x},\mathbf{y}|H;D)} \rho(\mathbf{z})\mathrm{d}\mathbf{z}\right] \\ &= \tilde{E}_{\mathcal{H}(D)}[\bar{\rho}(R(\mathbf{x}, \mathbf{y}|H; D))]\prod_{z\in\mathbf{z}} E_{\mathcal{H}(D)}[L(z)], \\ &= \tilde{E}_{\mathcal{H}(D)}[\bar{\rho}(R(\mathbf{x}, \mathbf{y}|H; D))]C_R\ell_p(\mathbf{x})^q, \end{aligned}$$

where C_R is a constant.

By substituting the above to Eq. (10), we derive the following $\ell_p(\mathbf{x})$:

$$\ell_p(\mathbf{x}) = \left\{\frac{\tilde{C}_R\tilde{m}_e(\mathbf{x})}{C_R\tilde{E}_{\mathcal{H}(D)}[\bar{\rho}(R(\mathbf{x}, \mathbf{y}|H, D))]}\right\}^{1/\tilde{q}}$$

This result and Eq. (12) provide the following $\delta P(V(\mathbf{x}, \mathbf{y}))$.

$$\begin{aligned} \delta P(V(\mathbf{x}, \mathbf{y})) &= \int_{V(\mathbf{x},\mathbf{y})} \rho(\mathbf{z})\mathrm{d}\mathbf{z} = \bar{\rho}(V(\mathbf{x}, \mathbf{y}))C_V\ell_p(\mathbf{x})\delta\Omega \\ &= C_V\left(\frac{\tilde{C}_R}{C_R}\right)^{1/\tilde{q}} \frac{\bar{\rho}(V(\mathbf{x}, \mathbf{y}))}{\tilde{E}_{\mathcal{H}(D)}[\bar{\rho}(R(\mathbf{x}, \mathbf{y}|H, D))]^{1/\tilde{q}}}\tilde{m}_e(\mathbf{x})^{1/\tilde{q}}\delta\Omega, \end{aligned} \tag{L1}$$

where $C_V\ell_p(\mathbf{x})\delta\Omega$ is the volume of $V(\mathbf{x}, \mathbf{y})$.

Now, for $\mathbf{y}$ on the edge of $R(\mathbf{x})$ belonging to $R^i_A(\mathbf{x})$, $\tilde{E}_{\mathcal{H}(D)}[\bar{\rho}(R(\mathbf{x}, \mathbf{y}|H; D))] = \bar{\rho}(V(\mathbf{x}, \mathbf{y}))$; since these densities are equal to the uniform density $\rho_A(\mathbf{x})$ in $\mathcal{X}_A$. Thus, Eq. (L1) becomes as follows:

$$\delta P(V(\mathbf{x}, \mathbf{y})) = C_V\left(\frac{\tilde{C}_R}{C_R}\right)^{1/\tilde{q}} \frac{\rho_A(\mathbf{x})}{\rho_A(\mathbf{x})^{1/\tilde{q}}}\tilde{m}_e(\mathbf{x})^{1/\tilde{q}}\delta\Omega.$$

Because the r.h.s. except its $\delta\Omega$ is constant, and $R^i_A(\mathbf{x})$ covers one half of the total solid angle of the point $\mathbf{x}$, we derive the following probability mass in $R^i_A(\mathbf{x})$:

$$\begin{aligned} P(R^i_A(\mathbf{x})) &= \int_{U^q/2} \mathrm{d}P(V(\mathbf{x}, \mathbf{y})) \\ &= \int_{U^q/2} C_V\left(\frac{\tilde{C}_R}{C_R}\right)^{1/\tilde{q}} \frac{\rho_A(\mathbf{x})}{\rho_A(\mathbf{x})^{1/\tilde{q}}}\tilde{m}_e(\mathbf{x})^{1/\tilde{q}}\mathrm{d}\Omega \\ &= \frac{U^q}{2}C_V\left(\frac{\tilde{C}_R}{C_R}\right)^{1/\tilde{q}} \frac{\rho_A(\mathbf{x})}{\rho_A(\mathbf{x})^{1/\tilde{q}}}\tilde{m}_e(\mathbf{x})^{1/\tilde{q}}, \end{aligned} \tag{L2}$$

where U^q is the total solid angle of the q-dimensional space Γ.

Next, we consider $\mathbf{y} \in G(\mathbf{x})$, *i.e.*, the edge of $R(\mathbf{x})$ belonging to $R^o_A(\mathbf{x}) \cup R_B(\mathbf{X})$. Because the densities in $\mathcal{X}_A$ and $\mathcal{X}_B$, *i.e.*, $\rho_A(\mathbf{x})$ and $\rho_B(\mathbf{x})$, are different, $\bar{\rho}(V(\mathbf{x}, \mathbf{y}))$ and $\tilde{E}_{\mathcal{H}(D)}[\bar{\rho}(R(\mathbf{x}, \mathbf{y}|H; D))]$ are between $\rho_A(\mathbf{x})$ and $\rho_B(\mathbf{x})$.

Thus, $\bar{\rho}(V(\mathbf{x}, \mathbf{y}))/\tilde{E}_{\mathcal{H}(D)}[\bar{\rho}(R(\mathbf{x}, \mathbf{y}|H; D))]^{1/\tilde{q}}$ can vary over $\mathbf{y} \in G(\mathbf{x})$.

Therefore, $\delta P(V(\mathbf{x},\mathbf{y}))$ in Eq. (L1) is upper-bounded as

$$\delta P(V(\mathbf{x},\mathbf{y})) \leq C_V \left(\frac{\tilde{C}_R}{C_R}\right)^{1/\tilde{q}} \tilde{\phi}(\mathbf{x})\tilde{m}_e(\mathbf{x})^{1/\tilde{q}}\delta\Omega$$

by

$$\tilde{\phi}(\mathbf{x}) = \sup_{\mathbf{y}\in G(\mathbf{x})} \frac{\bar{\rho}(V(\mathbf{x},\mathbf{y}))}{E_{\mathcal{H}}[\bar{\rho}(R(\mathbf{x},\mathbf{y}|H,D))]^{1/\tilde{q}}}$$

for any Ω.

Accordingly,

$$\begin{aligned} P(R_A^o(\mathbf{x})) + P(R_B(\mathbf{x})) = P(R_A^o(\mathbf{x}) \cup R_B(\mathbf{x})) &= \int_{U^q/2} \mathrm{d}P(V(\mathbf{x},\mathbf{y})) \\ &\leq \int_{U^q/2} C_V \left(\frac{\tilde{C}_R}{C_R}\right)^{1/\tilde{q}} \tilde{\phi}(\mathbf{x})\tilde{m}_e(\mathbf{x})^{1/\tilde{q}}\mathrm{d}\Omega \\ &= \frac{U^q}{2} C_V \left(\frac{\tilde{C}_R}{C_R}\right)^{1/\tilde{q}} \tilde{\phi}(\mathbf{x})\tilde{m}_e(\mathbf{x})^{1/\tilde{q}}. \qquad \text{(L3)} \end{aligned}$$

The ratio of Eqs. (L3) and (L2) provides this lemma. □

Because $k_A(\mathbf{x})$ follows a binomial distribution $\mathrm{B}(k, pr)$ where $pr = P(R_A(\mathbf{x}))/(P(R_A(\mathbf{x})) + P(R_B(\mathbf{x})))$; $P(R_B(\mathbf{x})) < P(R_A(\mathbf{x}))$, *i.e.*, $pr > 0.5$, gives $P(k_A(\mathbf{x}) > k/2) > 0.5$. Thus, $\mathbf{x} \in A$ is most probably classified to the correct class A. This fact and Lemma 1 provide the following required condition of $\phi(\mathbf{x})$ for the correct classification.

Lemma 2 *Given a border point* $\mathbf{x}$ *located in* $\mathcal{X}_A$ *under* $LMN_k(\mathbf{x})$, $\phi(\mathbf{x})$ *is required to satisfy the following inequality for* $\mathbf{x}$ *to be most probably classified correctly.*

$$\frac{\phi(\mathbf{x}) - 1}{2} < \frac{P(R_A^o(\mathbf{x}))}{P(R_A^i(\mathbf{x}))}.$$

Proof From Lemma 1, there exists $\phi(\mathbf{x})$ such that

$$\frac{P(R_A^o(\mathbf{x})) + P(R_B(\mathbf{X}))}{P(R_A^i(\mathbf{x}))} \leq \phi(\mathbf{x}) \Rightarrow P(R_B(\mathbf{X})) \leq \phi(\mathbf{x})P(R_A^i(\mathbf{x})) - P(R_A^o(\mathbf{x})).$$

The requirement $P(R_B(\mathbf{x})) < P(R_A(\mathbf{x}))$ for the most probable correct classification is satisfied, if the following condition is met.

$$\begin{aligned} &\phi(\mathbf{x})P(R_A^i(\mathbf{x})) - P(R_A^o(\mathbf{x})) < P(R_A(\mathbf{x})) = P(R_A^i(\mathbf{x})) + P(R_A^o(\mathbf{x})) \\ &\Rightarrow \frac{\phi(\mathbf{x}) - 1}{2} < \frac{P(R_A^o(\mathbf{x}))}{P(R_A^i(\mathbf{x}))}. \end{aligned}$$

□

This lemma indicates that the classification error of a border point $\mathbf{x}$ is most probably zero if $\phi(\mathbf{x})$ is less than 1, because $P(R_A^o(\mathbf{x}))/P(R_A^i(\mathbf{x}))$ is always positive. Moreover, the error is most probably very small even if $\phi(\mathbf{x})$ is more than but close to 1, since only a limited number of points $\mathbf{x} \in D$ having a short distance from the border are to fulfill $P(R_A^o(\mathbf{x})) \ll P(R_A^i(\mathbf{x}))$. Only these few points (if they exist) break the inequality of this lemma.

Based on these two lemmas, Theorem 2 characterises the accuracy of the kLMN classifier.

Proof of Theorem 2. The proof is provided in three parts corresponding to (i), (ii) and (iii).
(i) If $\tilde{q} = 1$, $\phi(\mathbf{x})$ defined in Lemma 1 is always almost 1, given the assumption that $\bar{\rho}(V(\mathbf{x}, \mathbf{y})) \simeq \tilde{E}_{\mathcal{H}(D)}[\bar{\rho}(R(\mathbf{x}, \mathbf{y}|H; D))]$. Accordingly, the required condition for the most probably correct classification presented in Lemma 2 is almost fulfilled in both cases of $A = S$ $(B = T)$ and $A = T$ $(B = S)$. Because this argument applies to all kLMN border points in D, both ε_S and ε_T are most probably very small.

(ii) Let $A = S$ and $B = T$ in Lemma 1 and 2. Since $\tilde{E}_{\mathcal{H}}[\bar{\rho}(R(\mathbf{x}, \mathbf{y}|H, D))] \simeq \bar{\rho}(V(\mathbf{x}, \mathbf{y}))$ holds by the assumption, we can approximately represent them by a value $\bar{\rho}_{ST}(\mathbf{x}, \mathbf{y}) \in [\rho_S(\mathbf{x}), \rho_T(\mathbf{x})]$; and let $\bar{\rho}_{ST}(\mathbf{x}) = \sup_{\mathbf{y} \in G(\mathbf{x})} \bar{\rho}_{ST}(\mathbf{x}, \mathbf{y})$. If $\tilde{q} > 1$, $\phi(\mathbf{x})$ defined in Lemma 1 can be rewritten as follows by using $\bar{\rho}_{ST}(\mathbf{x}, \mathbf{y})$ and $\bar{\rho}_{ST}(\mathbf{x})$:

$$\begin{aligned}
\phi(\mathbf{x}) &= \sup_{\mathbf{y} \in G(\mathbf{x})} \frac{\bar{\rho}(V(\mathbf{x}, \mathbf{y}))}{\rho_S(\mathbf{x})} \left\{ \frac{\rho_S(\mathbf{x})}{\tilde{E}_{\mathcal{H}(D)}[\bar{\rho}(R(\mathbf{x}, \mathbf{y}|H; D))]} \right\}^{1/\tilde{q}} \\
&= \sup_{\mathbf{y} \in G(\mathbf{x})} \frac{\bar{\rho}_{ST}(\mathbf{x}, \mathbf{y})^{1-1/\tilde{q}}}{\rho_S(\mathbf{x})^{1-1/\tilde{q}}} \\
&= \frac{\bar{\rho}_{ST}(\mathbf{x})^{1-1/\tilde{q}}}{\rho_S(\mathbf{x})^{1-1/\tilde{q}}}.
\end{aligned}$$

Because $\bar{\rho}_{ST}(\mathbf{x}) \in [\rho_S(\mathbf{x}), \rho_T(\mathbf{x})]$, $\phi(\mathbf{x})$ is bounded as

$$\phi(\mathbf{x}) \in [1, \{\rho_T(\mathbf{x})/\rho_S(\mathbf{x})\}^{1-1/\tilde{q}}].$$

Let $\phi(\mathbf{x})$ be at its upper bound, i.e.,

$$\phi(\mathbf{x}) = \left\{ \frac{\rho_T(\mathbf{x})}{\rho_S(\mathbf{x})} \right\}^{1-1/\tilde{q}}.$$

Because $\rho_T(\mathbf{x}) > \rho_S(\mathbf{x})$ always holds, $\phi(\mathbf{x}) > 1$ holds at any point $\mathbf{x}$ in $\mathcal{X}_S$. Hence, any point $\mathbf{x}$ having small $P(R^o_S(\mathbf{x}))$ relative to $P(R^i_S(\mathbf{x}))$, which breaks the inequality in Lemma 2, is misclassified with a non-negligible probability.

In addition, $P(R^o_S(\mathbf{x}))$ must increase relative to $P(R^i_S(\mathbf{x}))$ in the inequality of Lemma 2, when the ratio $\{\rho_T(\mathbf{x})/\rho_S(\mathbf{x})\}^{1-1/\tilde{q}}$ increases. This implies that for $\mathbf{x}$ in $\mathcal{X}_S$ to be most probably classified correctly, it must be farther from the border as the ratio increases. In other words, any point closer to the border than $\mathbf{x}$ is misclassified with a non-negligible probability. Therefore, the number of points misclassified with a non-negligible probability in $\mathcal{X}_S$ is most probably increased with respect to $\{\rho_T(\mathbf{x})/\rho_S(\mathbf{x})\}^{1-1/\tilde{q}}$. Because this argument applies to all kLMN border points in D, ε_S is most probably non-negligible and increases with respect to $(\bar{\rho}_T/\bar{\rho}_S)^{1-1/\tilde{q}}$ when $\bar{\phi}$ is close to the upper bound in the interval $[1, (\bar{\rho}_T/\bar{\rho}_S)^{1-1/\tilde{q}}]$.

(iii) Let $A = T$ and $B = S$ in Lemmas 1 and 2. Similarly to (ii), if $\tilde{q} > 1$, $\phi(\mathbf{x})$ defined in Lemma 1 is rewritten as

$$\phi(\mathbf{x}) = \frac{\bar{\rho}_{ST}(\mathbf{x})^{1-1/\tilde{q}}}{\rho_T(\mathbf{x})^{1-1/\tilde{q}}}.$$

Because $\rho_T(\mathbf{x}) > \rho_S(\mathbf{x})$ always holds, $\phi(\mathbf{x}) < 1$ holds at any point $\mathbf{x}$ in $\mathcal{X}_T$. Hence, the inequality of Lemma 2 is always satisfied, and all points in $\mathcal{X}_T$ are correctly classified in high probability. Accordingly, ε_T is most probably zero. □

References

Amari, S.-I., & Wu, S. (1999). Improving support vector machine classifiers by modifying kernel functions. *Neural Network*, *12*(6), 783–789.

Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. *SIGMOD Record*, *28*(2), 49–60.

Aryal, S., Ting, K. M., Haffari, G., & Washio, T. (2014a). m_p-dissimilarity: A data dependent dissimilarity measure. In *Proceedings of the IEEE international conference on data mining* (pp. 707–712).

Aryal, S., Ting, K. M., Wells, J. R., & Washio, T. (2014b). Improving iforest with relative mass. In *Advances in knowledge discovery and data mining* (pp. 510–521). Springer.

Borg, I., Groenen, P. J. F., & Mair, P. (2012). *Applied multidimensional scaling*. Berlin: Springer.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Chen, B., Ting, K. M., Washio, T., & Haffari, G. (2015). Half-space mass: A maximally robust and efficient data depth method. *Machine Learning*, *100*(2–3), 677–699.

Chen, Y., Garcia, E. K., Gupta, M. R., Rahimi, A., & Cazzanti, L. (2009). Similarity-based classification: Concepts and algorithms. *Journal Machine Learning Research*, *10*, 747–776.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Ertöz, L., Steinbach, M., & Kumar, V. (2003). Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the SIAM data mining conference* (pp. 47–58).

Ester, M., Kriegel, H-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery and data mining* (pp. 226–231).

Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). San Diego, CA: Academic Press Professional Inc.

Fukunaga, K., & Flick, T. E. (1984). An optimal global nearest neighbor metric. *IEEE Transactions on Pattern Analysis Machine Intelligence*, *6*(3), 314–318.

Guo, R., & Chakraborty, S. (2010). Bayesian adaptive nearest neighbor. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *3*(2), 92–105. https://doi.org/10.1002/sam.10067.

Hastie, T., & Tibshirani, R. (1996). Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(6), 607–616.

Holmes, C. C., & Adams, N. M. (2002). A probabilistic nearest neighbour method for statistical pattern recognition. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *64*(2), 295–306.

Jacobs, D. W., Weinshall, D., & Gdalyahu, Y. (2000). Classification with nonmetric distances: Image retrieval and class representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(6), 583–600.

Jarvis, R. A., & Patrick, E. A. (1973). Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers*, *100*(11), 1025–1034.

Krumhansl, C. L. (1978). Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density. *Psychological Review*, *85*(5), 445–463.

Lichman, M. (2013). UCI machine learning repository. Retrieved June 30, 2018 from http://archive.ics.uci.edu/ml.

Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the fifteenth international conference on machine learning* (pp. 296–304), San Francisco, CA, USA, Morgan Kaufmann.

Liu, F. T., Ting, K. M., & Zhou, Z-H. (2008). Isolation forest. In *Proceedings of the IEEE international conference on data mining* (pp. 413–422).

Liu, R. Y., Parelius, J. M., & Singh, K. (1999). Multivariate analysis by data depth: Descriptive statistics, graphics and inference. *The Annals of Statistics*, *27*(3), 783–840.

Liu, Y., Li, Z., Xiong, H., Gao, X., Junjie, W., & Sen, W. (2013). Understanding and enhancement of internal clustering validation measures. *IEEE Transactions on Cybernetics*, *43*(3), 982–994.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the national institute of sciences of India*, *2*, pp. 49–55.

Mosler, K. (2013). Depth statistics. In C. Becker, R. Fried, & S. Kuhnt (Eds.), *Robustness and complex data structures: Festschrift in Honour of Ursula Gather* (pp. 17–34). Berlin: Springer.

Mu, X., Ting, K. M., & Zhou, Z.-H. (2017). Classification under streaming emerging new classes: A solution using completely-random trees. *IEEE Transactions on Knowledge and Data Engineering*, *29*(8), 1605–1618.

Schleif, F.-M., & Tino, P. (2015). Indefinite proximity learning: A review. *Neural Computation*, *27*(10), 2039–2096.

Short, R. D., & Fukunaga, K. (1981). The optimal distance measure for nearest neighbor classification. *IEEE Transactions on Information Theory*, *27*(5), 622–627.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis* (Vol. 26). Boca Raton: CRC Press.

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining* (1st ed.). Boston, MA: Addison-Wesley Longman Publishing.

Tan, S. C., Ting, K. M., & Liu, T. F. (2011). Fast anomaly detection for streaming data. In *Proceedings of the twenty-second international joint conference on artificial intelligence* (pp. 1511–1516). AAAI Press.

Tan, X., Chen, S., Zhou, Z.-H., & Liu, J. (2009). Face recognition under occlusions and variant expressions with partial similarity. *IEEE Transactions on Information Forensics and Security*, *4*(2), 217–230.

Tang, B., & He, H. (2015). ENN: Extended nearest neighbor method for pattern recognition. *IEEE Computational Intelligence Magazine*, *10*(3), 52–60.

Ting, K. M., Washio, T., Wells, J. R., Liu, F. T., & Aryal, S. (2013a). DEMass: A new density estimator for big data. *Knowledge and Information Systems*, *35*(3), 493–524.

Ting, K. M., & Wells, J. R. (2010). Multi-dimensional mass estimation and mass-based clustering. In *Proceedings of the IEEE international conference on data mining* (pp. 511–520).

Ting, K. M., Zhou, G.-T., Liu, F. T., & Tan, J. S. C. (2010). Mass estimation and its applications. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining*, New York, NY, USA (pp. 989–998). ACM.

Ting, K. M., Zhou, G.-T., Liu, F. T., & Tan, S. C. (2013b). Mass estimation. *Machine Learning*, *90*(1), 127–160.

Ting, K. M., Zhu, Y., Carman, M., Zhu, Y., & Zhou, Z.-H. (2016). Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, New York, NY, USA (pp. 1205–1214). ACM.

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*(4), 327–352.

Vinh, N. X., Epps, J., & Bailey, J. (2009). Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of the 26th Annual international conference on machine learning*, New York, NY, USA (pp. 1073–1080). ACM.

Wang, F., & Sun, J. (2015). Survey on distance metric learning and dimensionality reduction in data mining. *Data Mining and Knowledge Discovery*, *29*(2), 534–564.

Weinberger, K. Q., & Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, *10*(2), 207–244.

Wells, J. R., Ting, K. M., & Washio, T. (2014). LiNearN: A new approach to nearest neighbour density estimator. *Pattern Recognition*, *47*(8), 2702–2720.

Wu, S., & Amari, S.-I. (2002). Conformal transformation of kernel functions: A data-dependent way to improve support vector machine classifiers. *Neural Processing Letters*, *15*(1), 59–67.

Xiong, H., Zhang, Y., & Chen, X.-W. (2007). Data-dependent kernel machines for microarray data classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *4*(4), 583–595.

Zadeh, P., Hosseini, R., & Sra, S. (2016). Geometric mean metric learning. In *Proceedings of the 33rd international conference on machine learning* (pp. 2464–2471).

Zhou, G.-T., Ting, K. M., Liu, F. T., & Yin, Y. (2012). Relevance feature mapping for content-based multimedia information retrieval. *Pattern Recognition*, *45*(4), 1707–1720.

Zhu, Y., Ting, K. M., & Carman, M. J. (2016). Density-ratio based clustering for discovering clusters with varying densities. *Pattern Recognition*, *60*, 983–997.

Affiliations

Kai Ming Ting[1] · **Ye Zhu**[2] · **Mark Carman**[3] · **Yue Zhu**[4] · **Takashi Washio**[5] · **Zhi-Hua Zhou**[4]

Kai Ming Ting
kaiming.ting@federation.edu.au

Mark Carman
mark.carman@monash.edu

Yue Zhu
zhuy@lamda.nju.edu.cn

Takashi Washio
washio@ar.sanken.osaka-u.ac.jp

Zhi-Hua Zhou
zhouzh@lamda.nju.edu.cn

[1] School of Engineering and Information Technology, Federation University, Churchill, Australia

[2] School of Information Technology, Deakin University, Geelong, Australia

[3] Faculty of Information Technology, Monash University, Melbourne, Australia

[4] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

[5] The Institute of Scientific and Industrial Research, Osaka University, Suita, Japan