Nearest-Neighbour-Induced Isolation Similarity and Its Impact on Density-Based Clustering

Xiaoyu Qin
Monash University
Victoria, Australia 3800
xiaoyu.qin@ieee.org

Kai Ming Ting
Federation University
Victoria, Australia 3842
kaiming.ting@federation.edu.au

Ye Zhu
Deakin University
Victoria, Australia 3125
ye.zhu@ieee.org

Vincent CS Lee
Monash University
Victoria, Australia 3800
vincent.cs.lee@monash.edu

Abstract

A recent proposal of data dependent similarity called Isolation Kernel/Similarity has enabled SVM to produce better classification accuracy. We identify shortcomings of using a tree method to implement Isolation Similarity; and propose a nearest neighbour method instead. We formally prove the characteristic of Isolation Similarity with the use of the proposed method. The impact of Isolation Similarity on density-based clustering is studied here. We show for the first time that the clustering performance of the classic density-based clustering algorithm DBSCAN can be significantly uplifted to surpass that of the recent density-peak clustering algorithm DP. This is achieved by simply replacing the distance measure with the proposed nearest-neighbour-induced Isolation Similarity in DBSCAN, leaving the rest of the procedure unchanged. A new type of clusters called mass-connected clusters is formally defined. We show that DBSCAN, which detects density-connected clusters, becomes one which detects mass-connected clusters, when the distance measure is replaced with the proposed similarity. We also provide the condition under which mass-connected clusters can be detected, while density-connected clusters cannot.

Introduction

Similarity measure is widely used in various data mining and machine learning tasks. In clustering analysis, its impact to the quality of result is critical(Steinbach, Ertöz, and Kumar 2004). A recent proposal of data dependent similarity called Isolation Kernel has enabled SVM to produce better classification accuracy by simply replacing the commonly used data independent kernel (such as Gaussian kernel) with Isolation Kernel (Ting, Zhu, and Zhou 2018). This is made possible on datasets of varied densities because Isolation Kernel is adaptive to local data distribution such that two points in a sparse region are more similar than two points of equal inter-point distance in a dense region. Despite this success, the kernel characteristic has not been formally proven yet.

This paper extends this line of research by investigating a different implementation of Isolation Similarity. We provide a formal proof of the characteristic of the Isolation Similarity for the first time since its introduction. In addition, we focus on using Isolation Similarity to improve the clustering performance of density-based clustering.

This paper identifies shortcomings of tree-based method currently employed in inducing Isolation Similarities (Ting, Zhu, and Zhou 2018). Instead, we investigate a different method to induce the data dependent Isolation Similarity, and evaluate its clustering performance using DBSCAN (Ester et al. 1996) in comparison with the two existing improvements of density-based clustering, i.e., DScale (Zhu, Ting, and Angelova 2018) and DP (Rodriguez and Laio 2014).

The rest of the paper is organised as follows. We reiterate Isolation Kernel, identify the shortcomings of using the tree-based method to induce Isolation Similarity, provide the proposed alternative that employs a nearest neighbour method, the lemma and proof of the characteristic of Isolation Similarity, and the investigation in using Isolation Similarity in density-based clustering.

The descriptions of existing works are framed in order to clearly differentiate from the contributions we made here.

Isolation Kernel

Isolation Kernel/Similarity is first proposed by (Ting, Zhu, and Zhou 2018) as a new similarity which can adapt to density structure of the given dataset, as opposed to commonly used data independent kernels such as Gaussian and Laplacian kernels.

In the classification context, Isolation Kernel has been shown to be an effective means to improve the accuracy of SVM, especially in datasets which have varied densities in the class overlap regions (Ting, Zhu, and Zhou 2018). This is achieved by simply replacing the commonly used data independent kernel such as Gaussian and Laplacian kernels with the Isolation Kernel.

In the context of SVM classifiers, Isolation Kernel (Ting, Zhu, and Zhou 2018) has been shown to be more effective than existing approaches such as distance metric learning (Zadeh, Hosseini, and Sra 2016; Wang and Sun 2015), multiple kernel learning (Rakotomamonjy et al. 2008; Gönen and Alpaydin 2011) and Random Forest kernel (Breiman 2000; Davies and Ghahramani 2014).

The characteristic of Isolation Kernel is akin to one aspect of human-judged similarity as discovered by psychologists (Krumhansl 1978; Tversky 1977), i.e., human will judge the two same Caucasians as less similar when compared in Europe (which have many Caucasians) than in Asia.

We restate the definition and kernel characteristic (Ting, Zhu, and Zhou 2018) below.

Let $D = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$ be a dataset sampled from an unknown probability density function $x_i \sim F$. Let $\mathcal{H}_\psi(D)$ denote the set of all partitions H that are admissible under D where each isolating partition $\theta \in H$ isolates one data point from the rest of the points in a random subset $\mathcal{D} \subset D$, and $|\mathcal{D}| = \psi$.

Definition 1 *For any two points $x, y \in \mathbb{R}^d$, Isolation Kernel of x and y wrt D is defined to be the expectation taken over the probability distribution on all partitioning $H \in \mathcal{H}_\psi(D)$ that both x and y fall into the same isolating partition $\theta \in H$:*

$$K_\psi(x, y|D) = \mathbb{E}_{\mathcal{H}_\psi(D)}[\mathbb{I}(x, y \in \theta \mid \theta \in H)] \quad (1)$$

where $\mathbb{I}(B)$ is the indicator function which outputs 1 if B is true; otherwise, $\mathbb{I}(B) = 0$.

In practice, K_ψ is estimated from a finite number of partitionings $H_i \in \mathcal{H}_\psi(D), i = 1, \ldots, t$ as follows:

$$K_\psi(x, y|D) = \frac{1}{t}\sum_{i=1}^{t} \mathbb{I}(x, y \in \theta \mid \theta \in H_i) \quad (2)$$

The characteristic of Isolation Kernel is: **two points in a sparse region are more similar than two points of equal inter-point distance in a dense region**, i.e.,

Characteristic of K_ψ: $\forall x, y \in \mathcal{X}_{\mathsf{S}}$ and $\forall x', y' \in \mathcal{X}_{\mathsf{T}}$ such that $\| x - y \| = \| x' - y' \|$, K_ψ satisfies the following condition:

$$K_\psi(x, y) > K_\psi(x', y') \quad (3)$$

where $\mathcal{X}_{\mathsf{S}}$ and $\mathcal{X}_{\mathsf{T}}$ are two subsets of points in sparse and dense regions of $\mathbb{R}^d$, respectively; and $\| x - y \|$ is the distance between x and y.

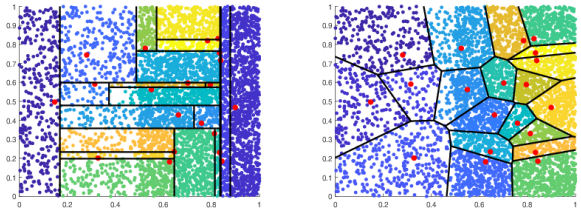
To get the above characteristic, the required property of the space partitioning mechanism is to create large partitions in the sparse region and small partitions in the dense region such that *two points are more likely to fall into a same partition in a sparse region than two points of equal inter-point distance in a dense region.*

Shortcomings of tree-based isolation partitioning

Isolation Kernel (Ting, Zhu, and Zhou 2018) employs isolation trees or iForest (Liu, Ting, and Zhou 2008) to measure the similarity of two points because its space partitioning mechanism produces the required partitions which have volumes that are monotonically decreasing wrt the density of the local region.

Here we identify two shortcomings in using isolation trees to measure Isolation Similarity, i.e., each isolation tree (i) employs axis-parallel splits; and (ii) is an imbalanced tree.

Figure 1(a) shows an example partitioning due to axis-parallel splits of an isolation tree. The tree-based isolating partitions generally satisfy the requirement of small partitions in dense region and large partitions in sparse region.



(a) Axis-parallel splitting (b) NN partitioning

Figure 1: Examples of two isolation partitioning mechanisms: Axis-parallel versus nearest neighbour (NN). On a dataset having two (uniform) densities, i.e., the right half has a higher density than the left half.

However, it produced some undesirable effect, i.e., some partitions are always overextended for the first few splits close to the root of an imbalanced tree[1]. These are manifested as elongated rectangles in Figure 1(a).

While using balanced trees can be expected to overcome this problem, the restriction to hyper-rectangles remains due to the use of axis-parallel splits.

To overcome these shortcomings of isolation trees, we propose to use a nearest neighbour partitioning mechanism which creates a Voronoi diagram (Aurenhammer 1991) where each cell is an isolating partition (i.e., isolating one point from the rest of the points in the given sample.) An example is provided in Figure 1(b). Note that these partitions also satisfy the requirement of small partitions in the dense region and large partitions in the sparse region. But they do not have the undesirable effect of elongated rectangles. We provide our implementation in the next section.

Nearest neighbour-induced Isolation Similarity

Instead of using trees in its first implementation (Ting, Zhu, and Zhou 2018), we propose to implement Isolation Similarity using nearest neighbours.

Like the tree method, the nearest neighbour method also produces each H model which consists of ψ isolating partitions θ, given a subsample of ψ points. Rather than representing each isolating partition as a hyper-rectangle, it is represented as a cell in a Voronoi diagram (Aurenhammer 1991), where the boundary between two points is the equal distance from these two points.

While the Voronoi diagram is nothing new, its use in measuring similarity is new.

Using the same notations as used earlier, H is now a Voronoi diagram, built by employing ψ points in $\mathcal{D}$, where each isolating partition or Voronoi cell $\theta \in H$ isolates one

[1]Imbalanced trees are a necessary characteristic of isolation trees (Liu, Ting, and Zhou 2008) for their intended purpose of detecting anomalies, where anomalies are expected to be isolated with few splits; and normal points can only be isolated using a large number of splits.

data point from the rest of the points in $\mathcal{D}$. We call the point which determines a cell as the cell centre.

Given a Voronoi diagram H constructed from a sample $\mathcal{D}$ of ψ points, the Voronoi cell centred at $z \in \mathcal{D}$ is:

$$\theta[z] = \{x \in \mathbb{R}^d \mid z = \underset{\mathsf{z} \in \mathcal{D}}{\operatorname{argmin}}\, \ell_p(x - \mathsf{z})\}.$$

where $\ell_p(x, y)$ is a distance function and we use $p = 2$ as Euclidean distance in this paper.

Definition 2 *For any two points $x, y \in \mathbb{R}^d$, the nearest neighbour-induced Isolation Similarity of x and y wrt D is defined to be the expectation taken over the probability distribution on all Voronoi diagrams $H \in \mathcal{H}_\psi(D)$ that both x and y fall into the same Voronoi cell $\theta \in H$:*

$$\begin{aligned} K_\psi(x, y \mid D) &= \mathbb{E}_{\mathcal{H}_\psi(D)}[\mathbb{I}(x, y \in \theta[z] \mid \theta[z] \in H)] \\ &= \mathbb{E}_{\mathcal{D} \sim D}[\mathbb{I}(x, y \in \theta[z] \mid z \in \mathcal{D})] \\ &= P(x, y \in \theta[z] \mid z \in \mathcal{D} \subset D) \end{aligned} \quad (4)$$

where P denotes the probability.

The Voronoi diagram has the required property of the space partitioning mechanism to produce large partitions in a sparse region and small partitions in a dense region. This yields the characteristic of Isolation Similarity : **two points in a sparse region are more similar than two points of equal inter-point distance in a dense region**.

The use of nearest neighbours facilitates a proof of the above characteristic that was previously hampered by the use of trees. We provide the proof in the next section.

Lemma and Proof of the characteristic of Isolation Similarity

Let $\rho(x)$ denote the density at point x, a lemma based on definition 4 is given below:

Lemma 1 *$\forall x, y \in \mathcal{X}_\mathsf{S}$ (sparse region) and $\forall x', y' \in \mathcal{X}_\mathsf{T}$ (dense region) such that $\forall_{z \in \mathcal{X}_\mathsf{S}, z' \in \mathcal{X}_\mathsf{T}}\, \rho(z) < \rho(z')$, the nearest neighbour-induced Isolation Similarity K_ψ has the characteristic that for $\ell_p(x - y) = \ell_p(x' - y')$ implies*

$$P(x, y \in \theta[z]) > P(x', y' \in \theta[z']) \equiv K_\psi(x, y \mid D) > K_\psi(x', y' \mid D)$$

Sketch of the proof: (i) If two points fall into the same Voronoi cell, then the distances of these individual points to this cell centre must be shorter than those to every other cell centre (or at most equal to those to one other cell centre) in a Voronoi diagram formed by all these cell centres. (ii) In a subset of ψ points, sampled from D, used to form a Voronoi diagram, the probability of two points falling into the same Voronoi cell can then be estimated based on the condition stated in (i). (iii) The probability of two points of equal inter-point distance falling into the same Voronoi cell is a monotonically decreasing function wrt the density of the cell.

PROOF 1 *Let a local region $V(x, y)$ covering both x and y as a ball centred at the middle between x and y having $\ell_p(x, y)$ as the diameter of the ball. Assume that the density in $V(x, y)$ is uniform and denoted as $\rho(V(x, y))$.*

Let $\mathcal{N}_\epsilon(x)$ be the ϵ-neighbourhood of x, i.e., $\mathcal{N}_\epsilon(x) = \{y \in D \mid \ell_p(x, y) \leqslant \epsilon\}$. The probability of both x and y are in the same Voronoi cell $\theta[z]$ is equivalent to the probability of a point $z \in \mathcal{D}$ being the nearest neighbour of both x and y wrt all other points in $\mathcal{D}$, i.e., the probability of selecting $\psi - 1$ points which are all located outside the region $U(x, y, z)$, where $U(x, y, z) = \mathcal{N}_{\ell_p(x,z)}(x) \cup \mathcal{N}_{\ell_p(y,z)}(y)$.

To simplify notation, $z \in \mathcal{D}$ is omitted. Then the probability of $x, y \in \theta[z]$ can be expressed as follows:

$$\begin{aligned} &P(x, y \in \theta[z] \mid z \in V(x, y)) \\ &\quad = P(z_1, z_2, \ldots, z_{(\psi-1)} \notin U(x, y, z)) \\ &\quad \propto (1 - \mathbb{E}_{z \sim V(x,y)}[|U(x, y, z)|]/|D|)^{(\psi-1)} \end{aligned}$$

where $|W|$ denotes the cardinality of W.

Assume that $U(x, y, z)$ is also uniformly distributed, having the same density $\rho(V(x, y))$, the expected value of $|U(x, y, z)|$ can be estimated as:

$$\begin{aligned} &\mathbb{E}_{z \sim V(x,y)}[|U(x, y, z)|] \\ &\quad = \mathbb{E}_{z \sim V(x,y)}[\upsilon(U(x, y, z)) \times \rho(V(x, y))] \\ &\quad = \mathbb{E}_{z \sim V(x,y)}[\upsilon(U(x, y, z))] \times \rho(V(x, y)) \end{aligned}$$

where $\upsilon(W)$ denotes the volume of W.

Thus, we have

$$P(x, y \in \theta[z] \mid z \in V(x, y)) \propto \left(1 - \mathbb{E}_{z \sim V(x,y)}[\upsilon(U(x, y, z))] \times \frac{\rho(V(x, y))}{|D|}\right)^{(\psi-1)} \quad (5)$$

In other words, the higher the density in the area around x and y, the smaller $P(x, y \in \theta[z] \mid z \in V(x, y))$ is, as the volume of $V(x, y)$ is constant given x and y.

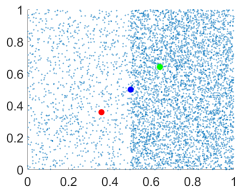
Given two pairs of points from two different regions but of equal interpoint distance as follows: $\forall x, y \in \mathcal{X}_\mathsf{S}$ (sparse region) and $\forall x', y' \in \mathcal{X}_\mathsf{T}$ (dense region) such that $\ell_p(x, y) = \ell_p(x', y')$.

Assume that data are uniformly distributed in both regions, and we sample $z, z' \in \mathcal{D}$ from D such that $z \in V(x, y)$ and $z' \in V(x', y')$. We have $\mathbb{E}_{z \sim V(x,y)}[\upsilon(U(x, y, z))] = \mathbb{E}_{z' \sim V(x',y')}[\upsilon(U(x', y', z'))]$ because the volume of $V(x, y)$ is equal to that of $V(x', y')$ for $\ell_p(x, y) = \ell_p(x', y')$, independent of the density of the region.

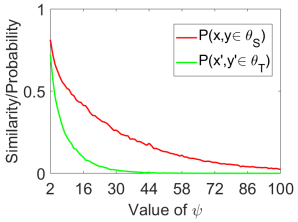
Supposing that we choose a sufficient large sample size ψ of $\mathcal{D}$ which contains points from both $V(x, y)$ and $V(x', y')$. When the data are uniformly distributed in $U(x, y, z) \in \mathcal{X}_\mathsf{S}$ and $U(x', y', z') \in \mathcal{X}_\mathsf{T}$, based on Equation 5, we have

$$\begin{aligned} &P(x, y \in \theta[z] \mid z \in V(x, y)) > \\ &\qquad P(x', y' \in \theta[z'] \mid z' \in V(x', y')) \\ &\equiv K_\psi(x, y \mid D) > K_\psi(x, y \mid D) \end{aligned}$$

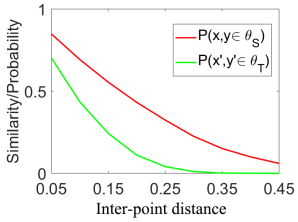
This means that x' and y' (in a dense region) are more like to be in different cells than x and y in $V(x, y)$ (in a sparse region), as shown in Figure 1. □



(a) Two regions of different densities



(b) ψ increases
Inter-point distance=0.2



(c) ψ=15
Inter-point distance increases

Figure 2: (a) Reference points used in the simulations, where inter-point distance $\| x - y \| = \| x' - y' \|$ increases. Simulation results as ψ increases (b); and as inter-point distance increases (c). $t = 10000$ is used.

A simulation validating the above analysis is given in Figure 2. It compares $P(x, y \in \theta_S)$ and $P(x', y' \in \theta_T)$ when x, y from a sparse region and x', y' from a dense region with equal inter-point distance. Given a fixed $\psi < |D|$ or a fixed inter-point distance, properties observed from Figure 2 are given as follows:

1. $P(x, y \in \theta_S) > P(x', y' \in \theta_T)$.
2. The rate of decrease of $P(x', y' \in \theta_T)$ is faster than that of $P(x, y \in \theta_S)$. Thus $P(x', y' \in \theta_T)$ reaches 0 earlier.

Isolation Dissimilarity and contour maps

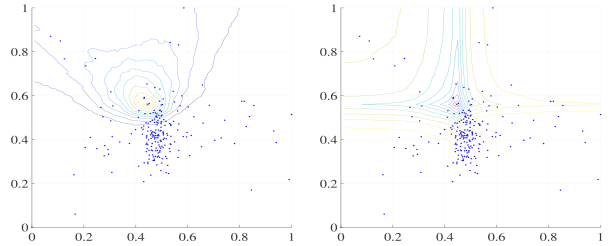
To be consistent with the concept of distance as a kind of dissimilarity, we use Isolation Dissimilarity hereafter:

Isolation dissimilarity: $\mathfrak{p}_\iota(x, y) = 1 - K_\psi(x, y)$.

Like ℓ_p norm, $\forall x, \mathfrak{p}_\iota(x, x) = 0$ and $\mathfrak{p}_\iota(x, y) = \mathfrak{p}_\iota(y, x)$. However, $\forall x \neq y$, $\mathfrak{p}_\iota(x, y)$ depends on the data distribution and how $\mathfrak{p}_\iota$ is implemented, not the geometric positions only.

We denote the nearest-neighbour-induced Isolation Dissimilarity $\mathfrak{p}_\iota$-aNNE; and the tree-induced version $\mathfrak{p}_\iota$-iForest. An example comparison of the contour maps produced the two dissimilarities are given in Figure 3. Note that the contour maps of $\mathfrak{p}_\iota$ depend on the data distribution, whereas that of ℓ_2 is not. Also, comparing to ℓ_2, the other dissimilarity change slower in area far from the centre point and faster in area close to the centre point.

We examine the impact of Isolation Dissimilarity on density-based clustering in the rest of the paper. We describe the neighbourhood density function commonly used in the density-based clustering and its counterpart called neighbourhood mass function in the next section.



(a) $\mathfrak{p}_\iota$-aNNE (b) $\mathfrak{p}_\iota$-iForest

Figure 3: Contour plots of $\mathfrak{p}_\iota$ on the Thyroid dataset (mapped to 2 dimensions using MDS (Borg, Groenen, and Mair 2012)). $\psi = 14$ is used in aNNE and iForest.

Neighbourhood density and mass functions

Neighbourhood mass function (Ting et al. 2016) was first proposed as a way to model data in terms of mass distribution, analogous to the neighbourhood density function used in modelling data as a density distribution. The key difference is the measure ∂ used in the following function of x: $\#\{y \in D \mid \partial(x, y) \leq \text{cutoff}\}$, where the boundary of the region within which the points are counted is set by a user-specified constant cutoff. When a distance measure is used, it becomes a neighbourhood density function as the ball has a fixed volume when the cutoff is fixed. It is a neighbourhood mass function when a data dependent dissimilarity is used as the volume of the 'ball' varies depending on the local distribution even if the cutoff is fixed.

Mass-connected clusters

Here we define mass-connected clusters defined in terms of neighbourhood mass function:

$$M_\alpha(x) = \#\{y \in D \mid \mathfrak{p}_\iota(x, y) \leq \alpha\}$$

Definition 3 *Using an α-neighbourhood mass estimator $M_\alpha(x) = \#\{y \in D \mid \mathfrak{p}_\iota(x, y) \leq \alpha\}$, mass-connectivity with threshold τ between x_1 and x_p via a sequence of p unique points from D, i.e., $\{x_1, x_2, x_3, ..., x_p\}$ is denoted as $MConnect^\tau_\alpha(x_1, x_p)$, and it is defined as:*

$$\begin{aligned} &MConnect^\tau_\alpha(x_1, x_p) \leftrightarrow \\ &\quad [(\mathfrak{p}_\iota(x_1, x_2) \leq \alpha) \wedge ((M_\alpha(x_1) \geq \tau) \vee (M_\alpha(x_2) \geq \tau))] \\ &\quad \vee [\exists_{\{x_1, x_2, ..., x_p\}} (\forall_{i \in \{2,...,p\}} \mathfrak{p}_\iota(x_{i-1}, x_i) \leq \alpha) \\ &\qquad \wedge (\forall_{i \in \{2,...,p-1\}} M_\alpha(x_i) \geq \tau)] \end{aligned} \tag{6}$$

The second line denotes direct connectivity between two neighbouring points when $p = 2$. The last two lines denote transitive connectivity when $p > 2$.

Definition 4 *A mass-connected cluster $\widetilde{C}$, which has a mode $\mathbf{c} = \arg\max_{x \in \widetilde{C}} M_\alpha(x)$, is a maximal set of points that are mass-connected with its mode, i.e., $\widetilde{C} = \{x \in D \mid MConnect^\tau_\alpha(x, \mathbf{c})\}$.*

Note that density-connectivity and density-connected clusters are similarly defined in DBSCAN (Ester et al. 1996) when $M_\alpha = \#\{y \in D \mid \mathfrak{p}_\iota(x,y) \leq \alpha\}$ is replaced with $N_\epsilon = \#\{y \in D \mid \ell_p(x,y) \leq \epsilon\}$ in the above two definitions. In other words, DBSCAN (Ester et al. 1996) which uses N_ϵ detects density-connected clusters; whereas DBSCAN which uses M_α detects mass-connected clusters.

The only difference between a density-connected cluster and a mass-connected cluster is the dissimilarity measure used in Equation 6. We called the DBSCAN procedure which employs M_α: MBSCAN, since it detects mass-connected clusters rather than density-connected clusters.

Condition under which MBSCAN detects all mass-connected clusters

Let a valley between two cluster modes be the points having the minimum estimated $M_\alpha(\cdot)$, i.e., $\mathfrak{g}_{ij}$, along any path linking cluster modes $\mathbf{c}_i$ and $\mathbf{c}_j$. A path between two points $\mathbf{x}$ and $\mathbf{y}$ is non-cyclic linking a sequence of unique points starting with $\mathbf{x}$ and ending with $\mathbf{y}$ where adjacent points lie in each other's α-neighbourhood: $\mathfrak{p}_\iota(\cdot,\cdot) \leq \alpha$.

Because $\mathfrak{p}_\iota$ (unlike ℓ_2 used in N_ϵ) is adaptive to the density of local data distribution, it is possible to adjust ψ and α to yield an M_α distribution such that all valley-points have close enough small values, if there exist such ψ and α.

In other words, for some data distributions F, there exist some ψ and α such that the distribution of $M_\alpha(\cdot)$ satisfies the following condition:

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

where $\hat{\mathfrak{g}}_{ij}$ is the largest of the minimum estimated $M_\alpha(\cdot)$ along any path linking cluster modes $\mathbf{c}_i$ and $\mathbf{c}_j$.

In data distributions F, MBSCAN is able to detect all mass-connected clusters because a threshold τ can be used to breaks all paths between the modes by assigning regions with estimated $M_\alpha(\cdot)$ less than τ to noise, i.e.,

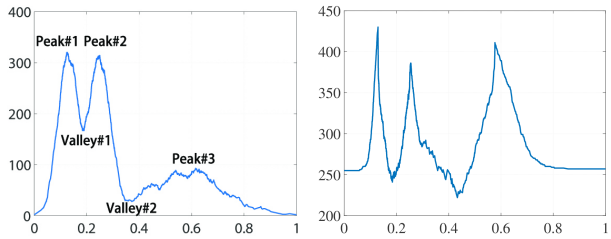
$$\exists_\tau \forall_{k, i \neq j \in \{1,\ldots,\aleph\}} M_\alpha(\mathbf{c}_k) \geqslant \tau > \hat{\mathfrak{g}}_{ij}$$

An example that F subsumes G, derived from the same dataset, is shown in Figure 4, where a hard distribution G in which DBSCAN fails to detect all clusters is shown in Figure 4(a); but MBSCAN succeeds [2].

In other words, the mass distribution afforded by M_α is more flexible than the density distribution generated by N_ϵ which leads directly to MBSCAN's enhanced cluster detection capability in comparison with DBSCAN, though both are using exactly the same algorithm, except the dissimilarity.

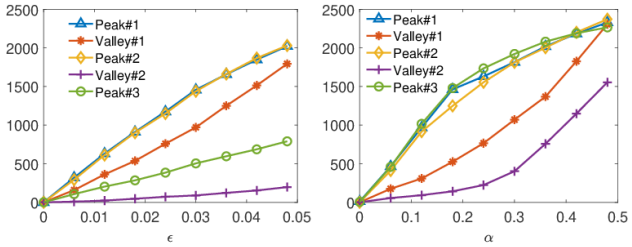
Figure 5 shows the change of neighbourhood function values wrt the change in their parameter for N_ϵ using ℓ_2 and M_α using $\mathfrak{p}_\iota$-aNNE. This example shows that no ϵ exists which enables DBSCAN to detect all three clusters. This is because the line for Peak#3 (which is the mode of the sparse cluster) has N_ϵ values in-between those of the two valleys. In contrast, many settings of α of M_α can be used to detect all three clusters because the lines of the two valleys are lower than those of the three peaks.

[2]Note that the above condition was first described in the context of using mass-based dissimilarity (Ting et al. 2018); but not in relation to mass-connected clusters. We have made the relation to mass-connected clusters more explicitly here.



(a) N_ϵ: ℓ_2 (b) M_α: $\mathfrak{p}_\iota$-aNNE

Figure 4: (a) A hard distribution for DBSCAN as estimated by N_ϵ, where DBSCAN (which uses N_ϵ) fails to detect all clusters using a threshold. (b) The distribution estimated by M_α from the same dataset, where MBSCAN (which uses M_α) succeeds in detecting all clusters using a threshold.



(a) N_ϵ vs ϵ (b) M_α vs α

Figure 5: Change of neighbourhood density/mass wrt its parameter. N_ϵ uses ℓ_2; and M_α uses $\mathfrak{p}_\iota$-aNNE. The peak numbers and valley numbers refer to those shown in Figure 4(a).

Experiments

The aim of the experiments is to compare the clustering performance of DBSCAN using different dissimilarities relative to that of the state-of-the-art density-based clustering algorithm DP (Rodriguez and Laio 2014). In addition to the three dissimilarity measures, i.e., ℓ_2, $\mathfrak{p}_\iota$-iForest and $\mathfrak{p}_\iota$-aNNE, two recent distance transformation method called ReScale (Zhu, Ting, and Carman 2016) and DScale (Zhu, Ting, and Angelova 2018) are also included. Note that DBSCAN using $\mathfrak{p}_\iota$ are denoted as MBSCAN, as they are mass-based clustering methods.

All algorithms used in our experiments are implemented in Matlab (the source code with demo can be obtained from https://github.com/cswords/anne-dbscan-demo). We produced the GPU accelerated versions of all implementations. The experiments ran on a machine having CPU: i5-8600k 4.30GHz processor, 8GB RAM; and GPU: GTX Titan X with 3072 1075MHz CUDA (Owens et al. 2008) cores & 12GB graphic memory.

A total of 20 datasets[3] are used in the experiments.

[3]The artificial datasets are from http://cs.uef.fi/sipu/datasets/

Table 1: Clustering results in F_1 scores. The best performer is boldfaced; the second best is underlined.

Datasets				DP	DBSCAN			MBSCAN	
Name	#Points	#Dim.	#Clusters	ℓ_2	ℓ_2	ReScale	DScale	iForest	aNNE
Artificial data			(average⇒)	*0.961*	*0.852*	*0.941*	*0.985*	*0.969*	*0.981*
aggregation	788	2	7	**1.000**	0.997	0.996	**1.000**	0.996	**1.000**
compound	399	2	6	0.867	0.791	0.862	**0.942**	0.875	0.918
jain	373	2	2	**1.000**	0.976	**1.000**	**1.000**	**1.000**	**1.000**
pathbased	300	2	3	0.943	0.828	0.864	0.987	0.986	**0.995**
hard distribution	1500	2	3	0.994	0.667	0.985	**0.995**	0.987	0.992
High-dimensional data			(average⇒)	*0.627*	*0.446*	*0.521*	*0.491*	*0.568*	*0.727*
ALLAML	72	7129	2	0.706	0.484	0.729	0.484	0.747	**0.820**
COIL20	1440	1024	20	0.724	0.842	0.861	0.839	0.865	**0.952**
Human Activity	1492	561	6	**0.595**	0.331	0.352	0.374	0.402	0.502
Isolet	1560	617	26	0.517	0.194	0.234	0.426	0.289	**0.605**
lung	203	3312	5	0.703	0.489	0.544	0.489	0.649	**0.921**
TOX 171	171	5748	4	0.519	0.336	0.403	0.336	0.454	**0.563**
General data			(average⇒)	*0.876*	*0.680*	*0.820*	*0.860*	*0.873*	*0.896*
breast	699	9	2	**0.970**	0.824	0.951	0.966	0.963	0.964
control	600	60	6	0.736	0.531	0.663	0.844	0.738	**0.854**
gps	163	6	2	0.811	0.753	0.811	0.811	**0.819**	0.766
iris	150	4	3	0.967	0.848	0.905	0.926	0.966	**0.973**
seeds	210	7	3	0.909	0.750	0.885	0.871	0.907	**0.922**
shape	160	17	9	0.761	0.581	0.680	0.722	0.725	**0.787**
thyroid	215	5	3	0.868	0.584	0.850	0.828	0.915	**0.916**
WDBC	569	30	2	**0.933**	0.600	0.765	0.894	0.895	0.927
wine	178	13	3	0.933	0.645	0.866	0.881	0.927	**0.959**
			Grand Average	0.823	0.653	0.760	0.781	0.805	0.867
			Number of datasets with the **Best** F_1 score	5	0	1	4	2	14
			#wins/#draws/#loses wrt MBSCAN-$\mathfrak{p}_\iota$-aNNE	5/2/13	0/0/20	1/2/18	4/2/14	1/1/18	-

They are from three categories: 5 artificial datasets, 6 high-dimensional datasets, and 9 general datasets. They are selected because they represent diverse datasets in terms of data size, number of dimensions and number of clusters. The data characteristics of these datasets are shown in the first four columns of Table 1. All datasets are normalised using the min-max normalisation so that each attribute is in [0,1] before the experiments begin.

We compared all clustering results in terms of the best F_1 score (Rijsbergen 1979) [4] that is obtained from a search of the algorithm's parameter. We search each parameter within a reasonable range. The ranges used for all algorithms/dissimilarities are provided in Table 2. Because $\mathfrak{p}_\iota$ used in MBSCAN is based on randomised methods, we report the mean F_1 score over 10 trials for each dataset.

Table 2: Search ranges of parameters used.

	Description	Candidates
DP	Target cluster number	$k \in [2...40]$
	neighbourhood size in N_ϵ	$\epsilon \in [0.001...0.999]$
DBSCAN	$MinPts$	$MinPts \in [2...40]$
MBSCAN	neighbourhood size in N_ϵ	$\epsilon \in [0.001...0.999]$
ReScale	precision factor	$f = 200$ *
DScale	neighbourhood size in N_η	$\eta \in [0.05...0.95]$
aNNE	Ensemble size	$t = 200$
iForest	Subsample size	$\psi \in [2, \lceil n/2 \rceil]$ †

* f parameter is required for ReScale only.
† A search of 10 values with equal interval in the range.

Clustering results

Table 1 shows that MBSCAN using $\mathfrak{p}_\iota$-aNNE has the best performance overall. Its F_1 scores are the best on 14 out of 20 datasets. The closest contender DP has the best F_1 scores on 5 datasets only. In two other performance measures, MBSCAN using $\mathfrak{p}_\iota$-aNNE has 13 wins, 2 draws and 5 losses

(Gionis, Mannila, and Tsaparas 2007; Zahn 1971; Chang and Yeung 2008; Jain and Law 2005) except that the hard distribution dataset is from https://sourceforge.net/p/density-ratio/ (Zhu, Ting, and Carman 2016), 5 high-dimensional data are from http://featureselection.asu.edu/datasets.php (Li et al. 2016), and the rest of the datasets are from http://archive.ics.uci.edu/ml (Dheeru and Karra Taniskidou 2017).

[4] $F_1 = \frac{1}{k}\sum_{i=1}^{k}\frac{2p_i r_i}{p_i + r_i}$, where p_i and r_i are the precision and the recall for cluster i, respectively. F_1 is preferred over other evaluation measures such as Purity (Manning, Raghavan, and Schütze 2008) and Normalized Mutual Information (NMI) (Strehl and Ghosh 2002) because these measures do not take into account noise points which are identified by a clustering algorithm. Based on these measures, algorithms can obtain a high clustering performance by assigning many points to noise, which can be misleading in a comparison.

against DP; and has higher average F_1 score too (0.867 versus 0.823).

One notable standout is on the high-dimensional datasets: MBSCAN using $\mathfrak{p}_i$-aNNE has the largest gap in average F_1 score in comparison with other contenders among the three categories of datasets. With reference to DBSCAN, the gap is close to 0.3 F_1 score; even compare with the closest contender DP, the gap is 0.1 F_1 score. The superiority of $\mathfrak{p}_i$-aNNE over $\mathfrak{p}_i$-iForest is also highlighted on these high-dimensional datasets.

MBSCAN using $\mathfrak{p}_i$-iForest wins over the original DBSCAN on all datasets except one (aggregation). This version of MBSCAN uplifted the clustering performance of DBSCAN significantly to almost the same level of DP.

A significance test is conducted over MBSCAN with $\mathfrak{p}_i$-aNNE, MBSCAN with $\mathfrak{p}_i$-iForest and DP. Figure 6 shows the result of the test—MBSCAN using $\mathfrak{p}_i$-aNNE performs the best and is significantly better than DP and MBSCAN using $\mathfrak{p}_i$-iForest; and there is no significant difference between DP and MBSCAN using $\mathfrak{p}_i$-iForest.

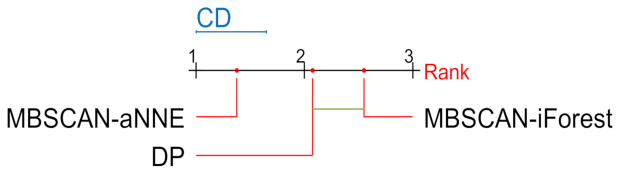


Figure 6: Critical difference (CD) diagram of the post-hoc Nemenyi test ($\alpha = 0.10$). A line is showed between two algorithms if their rank gap is smaller than CD; otherwise, the difference is significant.

DBSCAN is known to be sensitive to its parameter settings. As MBSCAN is using exactly the same algorithm, it has same sensitivity.

A caveat is in order. Although Isolation Similarity consistently outperforms distance measure in DBSCAN, our preliminary experiment using DP shows that the result is mixed. An analysis of DP, similar to that provided in this paper, is required in order to ascertain the condition(s) under which DP performs well and Isolation Similarity can help.

Complexity and runtime

MBSCAN with $\mathfrak{p}_i$-aNNE is much faster than MBSCAN with $\mathfrak{p}_i$-iForest because the time complexity to build a maximum-size isolation tree and testing one point is $O(\psi^2)$; and aNNE takes $O(\psi)$. The space complexity to store trained aNNE model is $O(t \cdot \psi)$; and that of iForest is $O(t \cdot \psi \cdot \log \psi)$.

Table 3 shows the GPU runtime results on the four largest datasets. In contrast to DBSCAN and DP, MBSCAN needs to pre-compute the dissimilarity matrix in the pre-processing step, and this takes $O(n^2)$ time. This pre-processing constitutes the most of the time of MBSCAN reported in Table 3. $\mathfrak{p}_i$-aNNE is still faster than ReScale in high-dimensional datasets, though it is one order of magnitude slower than DBSCAN and DP.

Table 3: Runtime in GPU seconds

Datasets	DP	DBSCAN		MBSCAN	
		Original	ReScale	iForest	aNNE
Hard dist.	0.11	0.07	0.08	154	0.50
COIL20	0.03	0.02	3.74	762	0.45
Human Act.	0.10	0.03	2.12	146	0.48
Isolet	0.08	0.03	2.64	472	0.48

Summary: aNNE versus iForest implementations of Isolation Similarity

We find that the aNNE implementation of Isolation Similarity is better than the iForest implementation because the aNNE implementation is more:

- Effective on datasets with varied densities and high-dimensional datasets.
- Amenable to GPU acceleration because aNNE can be implemented in almost pure matrix manipulations. Thus, aNNE runs many orders of magnitude faster than iForest if a large ψ is required because aNNE in the GPU implementation has almost constant runtime wrt ψ.
- Stable because aNNE's randomisation is a result of sampling data subsets only.

Conclusions

We make four contributions in this paper:

1) Identifying shortcomings of tree-induced Isolation Similarity; proposing a nearest neighbour-induced Isolation Similarity to overcome these shortcomings; and establishing three advantages of the nearest neighbour-induced Isolation Similarity over the tree-induced one.
2) Formally proving the characteristic of the nearest neighbour-induced Isolation Similarity. This is the first proof since the introduction of Isolation Kernel (Ting, Zhu, and Zhou 2018).
3) Providing a formal definition of mass-connected clusters and an explanation why detecting mass-connected clusters is a better approach in overcoming the shortcoming of DBSCAN (which detects density-connected clusters) in datasets with varied densities. This differs fundamentally from the existing density-based approaches of the original DBSCAN, DP and ReScale which all employ a distance measure to compute density.
4) Conducting an empirical evaluation to validate the advantages of (i) nearest-neighbour-induced Isolation Similarity over tree-induced Isolation Similarity; and (ii) mass-based clustering using Isolation Similarity over four density-based clustering algorithms, i.e., DBSCAN, DP, ReScale and DScale.

In addition, we show for the first time that it is possible to uplift the clustering performance of the classic DBSCAN, through the use of nearest-neighbour-induced Isolation Similarity, to surpass that of DP—the state-of-the-art density-based clustering algorithm.

Acknowledgements

This material is based upon work supported by eSolutions of Monash University (Xiaoyu Qin); and 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 (Kai Ming Ting).

References

Aurenhammer, F. 1991. Voronoi diagrams—A survey of a fundamental geometric data structure. *ACM Computing Surveys* 23(3):345–405.

Borg, I.; Groenen, P. J.; and Mair, P. 2012. *Applied multidimensional scaling*. Springer Science & Business Media.

Breiman, L. 2000. Some infinity theory for predictor ensembles. *Technical Report 577. Statistics Dept. UCB.*

Chang, H., and Yeung, D.-Y. 2008. Robust path-based spectral clustering. *Pattern Recognition* 41(1):191–203.

Davies, A., and Ghahramani, Z. 2014. The random forest kernel and creating other kernels for big data from random partitions. *arXiv:1402.4293*.

Dheeru, D., and Karra Taniskidou, E. 2017. UCI machine learning repository.

Ester, M.; Kriegel, H.-P.; Sander, J.; and Xu, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 96, 226–231.

Gionis, A.; Mannila, H.; and Tsaparas, P. 2007. Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1(1):4.

Gönen, M., and Alpaydin, E. 2011. Multiple kernel learning algorithms. *Journal Machine Learning Research* 12:2211–2268.

Jain, A. K., and Law, M. H. 2005. Data clustering: A user's dilemma. In *International conference on pattern recognition and machine intelligence*, 1–10. Springer.

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.

Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Robert, T.; Tang, J.; and Liu, H. 2016. Feature selection: A data perspective. *arXiv:1601.07996*.

Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2008. Isolation forest. In *The Eighth IEEE International Conference on Data Mining*, 413–422. IEEE.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

Owens, J. D.; Houston, M.; Luebke, D.; Green, S.; Stone, J. E.; and Phillips, J. C. 2008. Gpu computing. *Proceedings of the IEEE* 96(5):879–899.

Rakotomamonjy, A.; Bach, F. R.; Canu, S.; and Grandvalet, Y. 2008. SimpleMKL. *Journal of Machine Learning Research* 9(Nov):2491–2521.

Rijsbergen, C. J. V. 1979. *Information Retrieval*. Newton, MA, USA: Butterworth-Heinemann, 2nd edition.

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

Steinbach, M.; Ertöz, L.; and Kumar, V. 2004. The challenges of clustering high dimensional data. In *New directions in statistical physics*. Springer. 273–309.

Strehl, A., and Ghosh, J. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3(Dec):583–617.

Ting, K. M.; Zhu, Y.; Carman, M.; Zhu, Y.; and 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*, 1205–1214. ACM.

Ting, K. M.; Zhu, Y.; Carman, M.; Zhu, Y.; Washio, T.; and Zhou, Z.-H. 2018. Lowest probability mass neighbour algorithms: relaxing the metric constraint in distance-based neighbourhood algorithms. *Machine Learning,* doi.org/10.1007/s10994–018–5737–x.

Ting, K. M.; Zhu, Y.; and Zhou, Z.-H. 2018. Isolation kernel and its effect on SVM. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2329–2337. ACM.

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

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

Zadeh, P.; Hosseini, R.; and Sra, S. 2016. Geometric mean metric learning. In *International Conference on Machine Learning*, 2464–2471.

Zahn, C. T. 1971. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on computers* 100(1):68–86.

Zhu, Y.; Ting, K. M.; and Angelova, M. 2018. A distance scaling method to improve density-based clustering. In Phung, D.; Tseng, V. S.; Webb, G. I.; Ho, B.; Ganji, M.; and Rashidi, L., eds., *Advances in Knowledge Discovery and Data Mining*, 389–400. Cham: Springer International Publishing.

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