Pattern Recognition 137 (2023) 109287

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Overcoming weaknesses of density peak clustering using a data-dependent similarity measure

Zafaryab Rasool[1,*], Sunil Aryal[1], Mohamed Reda Bouadjenek, Richard Dazeley

School of Information Technology, Deakin University, Waurn Ponds Campus, Geelong, VIC 3216, Australia

ARTICLE INFO

Article history:
Received 24 December 2021
Revised 6 November 2022
Accepted 27 December 2022
Available online 30 December 2022

Keywords:
Clustering
Density peak clustering
Similarity measure
Data-dependent similarity

ABSTRACT

Density Peak Clustering (DPC) is a popular state-of-the-art clustering algorithm, which requires pairwise (dis)similarity of data objects to detect arbitrary shaped clusters. While it is shown to perform well for many applications, DPC remains: (i) not robust for datasets with clusters having different densities, and (ii) sensitive to the change in the units/scales used to represent data. These drawbacks are mainly due to the use of the data-independent similarity measure based on the Euclidean distance. In this paper, we address these issues by proposing an effective data-dependent similarity measure based on *Probability Mass*, which we call *MP-Similarity*, and by incorporating it in DPC to create MP-DPC, a data-dependent variant of DPC. We evaluate and compare MP-DPC against diverse baselines using several clustering metrics and datasets. Our experiments demonstrate that: (a) MP-DPC produces better clustering results than DPC using the Euclidean distance and existing data-dependent similarity measures; (b) MP-Similarity coupled with Shared-Nearest-Neighbor-based density metric in DPC further enhances the quality of clustering results; and (c) unlike DPC with existing data-independent and data-dependent similarity measures, MP-DPC is robust to the change in the units/scales used to represent data. Our findings suggest that MP-Similarity provides a more viable solution for DPC in datasets with unknown distribution or units/scales of features, which is often the case in many real-world applications.

1. Introduction

Clustering is a fundamental task in data mining used to find groups (or clusters) of similar objects in a dataset based on a notion of similarity. Clustering has applications in areas such as pattern recognition [1], bioinformatics [2], image segmentation [3], information retrieval [4], etc. Over the last decades, a large number of clustering techniques following different mechanisms have been developed to support different applications. In particular, state-of-the-art Density Peak Clustering (DPC) technique [5] has gained growing attention over the last few years due to its simplicity with fewer initial parameters and ability to detect clusters of arbitrary shapes.

DPC can find arbitrary shaped clusters using the distance-based notion of (dis)similarity of data objects, i.e., objects with small distances are more similar than objects with larger distances. The basic idea of DPC is that the cluster centres are characterised as objects with higher local density than their neighbourhood and are separated by relatively large distances from other objects of higher density. Based on this idea, DPC first defines two measures for each object: (1) its *local density* ρ and (2) its *distance* δ *to the nearest higher density neighbor*, and then, it selects the objects with high ρ and large δ as the cluster centers. Finally, it assigns the rest of the objects to the clusters containing their nearest higher density neighbors. Because of its simplicity, DPC has been used for many applications such as neuroscience [6], remote sensing [7], biology [8], video-segmentation [9], traffic-network [10], computer vision [11], text mining [12], etc.

However, despite its popularity and simplicity, we note that DPC has two major weaknesses. First, DPC may fail to correctly identify clusters with highly variable densities. Second, the clustering results of DPC are very sensitive to units/scales used to measure/represent data features. Indeed, in real-world examples, data can be measured and expressed in different forms. For example, sample variability can be measured in standard deviation (σ) or variance (σ^2) and people's ability to borrow can be expressed in terms of debt-to-income ratio or income-to-debt ratio (inverse of each other). These issues are mainly due to the use of distance as the notion of similarity between data objects as we later discuss in Section 3.3.

* Corresponding author.
E-mail addresses: zafaryab.rasool@deakin.edu.au (Z. Rasool), sunil.aryal@deakin.edu.au (S. Aryal), reda.bouadjenek@deakin.edu.au (M.R. Bouadjenek), richard.dazeley@deakin.edu.au (R. Dazeley).
[1] Zafaryab Rasool and Sunil Aryal contributed equally to this work.

https://doi.org/10.1016/j.patcog.2022.109287
0031-3203/

In this paper, we argue that the above-mentioned limitations of DPC can be addressed using a data-dependent similarity measure that is robust to units/scales of data representation. In particular, we propose a similarity measure based on probability mass called *MP-Similarity* that uses the m_0-dissimilarity [13], and then, we incorporate it into DPC to get a variant that we call MP-DPC. This latter uses the same core procedures of the original DPC algorithm while replacing the distance-based (dis)similarity measure with MP-Similarity. We also use MP-Similarity with the Shared-Nearest-Neighbor-based Density Peak Clustering algorithm (SNNDPC) [14], a variant of DPC. We call this new variant of SNNDPC as MP-SNNDPC. We show through our extensive experiments on a wide range of real datasets that MP-DPC and MP-SNNDPC can detect clusters with varying densities and that their clustering results are robust to changing data units/scales. We also demonstrate that they significantly outperform the basic DPC and SNNDPC algorithms which are based on the Euclidean distance.

2. Background

Over the past few decades, several clustering techniques have been developed to address different problems in diverse applications. Below, we briefly review the main clustering methods based on the approaches they adopt to grouping similar objects. For a comprehensive survey on the topic, we refer the reader to [15].

Partition-based methods construct several partitions of the data, where each partition is a cluster. These methods require prespecifying the number of clusters such as k in k-means [16], a popular representative algorithm of partition-based clustering. The k-means algorithm randomly initialises k objects as centers (or cluster means), and assigns the rest of the objects to the nearest center. It then performs the following two steps: (i) updating center of each cluster based on the current assignment and (ii) assigning objects to clusters containing the nearest centers. k-means alternate between these steps until it finds the best centers i.e., no further change occurs. A major limitation of this family of algorithms is that they are not suitable to detect arbitrary shaped clusters. Moreover, the clustering results are sensitive to the initial selection of the cluster centers.

Hierarchical-based methods produce a tree-based hierarchical representation of clusters with the root representing the entire set of objects as one cluster and pairs of clusters or singletons are located at the lower levels of the tree. Tree partitions are constructed based on a proximity/similarity matrix and a partition of this tree generates the clustering results. Single linkage hierarchical clustering [17] is a representative example of this family. It determines the distance between two closest objects in different clusters and merge the two clusters if they contain objects separated by shortest distance. Single linkage follows a bottom-up approach such that it starts with singletons and continues merging two clusters until all the objects belong to one cluster.

Density-based clustering methods assume that the clusters are located in dense regions in the space separated by regions of lower density and are suitable for detecting irregularly-shaped clusters. DBSCAN [18], a popular representative of density-based clustering, finds objects with density greater than a threshold, which are connected to form clusters. However, selecting an appropriate threshold is difficult. Density Peak Clustering (DPC) [5] is another popular algorithm of this family that we will detail in the next section.

In addition to those discussed above, there exist other clustering methods in the literature. For example, **Distribution-based clustering methods** assume that data objects in a cluster have more likelihood of belonging to the same distribution (e.g., Gaussian distribution). These algorithms work well when the distribution of the data is known beforehand. Expectation-maximization [19] is a popular example of this family. **Grid-based methods**, such as STING [20], partition the original data space into several grids which are linked to form clusters based on statistical information such as the mean and the variance of the data objects.

3. Density peak clustering (DPC)

In this section, we first present in details the DPC algorithm, then, we describe a few of its existing variants and extensions, and finally, we analyse its main drawbacks.

3.1. DPC Algorithm

DPC [5] is a clustering algorithm based on the observation that *cluster centres* are characterized by: (i) *locally higher density*, i.e., a cluster centre has a higher-density neighbourhood than its neighbouring objects, and (ii) *relatively large separation*, i.e., cluster centres are at relatively large distances from other objects with higher local densities. Based on the above idea, DPC defines two quantities for each object that are used to identify the cluster centers from the rest of the objects. Formally, given a dataset $\mathcal{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)}\}$ with m objects (i.e., instances), where each data object $\mathbf{x}^{(i)} \in \mathbb{R}^n$ is an n-dimensional input feature vector, DPC proceeds in the following steps:

1. Compute the local density $\rho_{\mathbf{x}}$ of each object $\mathbf{x}$ as follows:

$$\rho_{\mathbf{x}} = \sum_{\mathbf{y} \in \mathcal{D}} \chi\left(dist(\mathbf{x},\mathbf{y}) - d_c\right) \tag{1}$$

   where $dist(\mathbf{x},\mathbf{y})$ represents the distance between object $\mathbf{x}$ and $\mathbf{y}$, $\chi(z) = 1$ if $z < 0$ and 0 otherwise, and d_c is the cutoff distance. In other words, the local density $\rho_{\mathbf{x}}$ of $\mathbf{x}$ is the number of objects that lie within distance d_c from $\mathbf{x}$.
2. For each object $\mathbf{x}$, compute the distance $\delta_{\mathbf{x}}$ which is the minimum distance between $\mathbf{x}$ and any other object $\mathbf{y} \in \mathcal{D}$ with density higher than $\rho_{\mathbf{x}}$. Specifically, the distance $\delta_{\mathbf{x}}$ is computed as follows:

$$\delta_{\mathbf{x}} = \min_{\mathbf{y} \neq \mathbf{x} \wedge \rho_{\mathbf{y}} > \rho_{\mathbf{x}}} \{dist(\mathbf{x},\mathbf{y})\} \tag{2}$$

   Here, the object $\mathbf{y}$ is the *nearest neighbor of high density* for $\mathbf{x}$. However, for the object $\mathbf{x}$ with the highest density, the distance $\delta_{\mathbf{x}}$ is computed as follows: $\delta_{\mathbf{x}} = \max_{\mathbf{y}} \{dist(\mathbf{x},\mathbf{y})\}$.
3. Using the computed values ρ and δ values, DPC aims to distinguish the cluster centres (peaks) from the rest of the objects. In particular, an object with high local density has its nearest neighbour of higher density relatively far and therefore has a large δ. Based on this idea, cluster centres are set to objects with high ρ and anomalously large δ. For selecting the cluster centres, a quantity γ is computed for each object as follows:

$$\gamma_{\mathbf{x}} = \rho_{\mathbf{x}} \times \delta_{\mathbf{x}} \tag{3}$$

   Objects with large γ values are set to be cluster centres.
4. Finally, once the cluster centres have been identified, the rest of the objects are then assigned to the clusters containing their nearest neighbour of higher density. This step is performed directly as assignment uses the nearest neighbour of higher density information obtained during the computation of δ in Eq. (2).

3.2. Variants of DPC

Despite being a state-of-the-art clustering algorithm, DPC has limitations in dealing with clusters with complex shapes/structures such as varying densities [14]. To improve the performance of DPC algorithms in various applications, different variants of

DPC have been proposed in the literature. Most of these employ measures that rely on the K-Nearest Neighbour (KNN) information of objects to capture the neighbourhood information. Shared-Nearest-Neighbour-based density peak clustering (SNNDPC) [14] uses Shared Nearest Neighbours (SNN) as a means to compute the similarities between objects and it selects the cluster centers based on the nearest neighbours and shared neighbours information. Furthermore, a two-step allocation strategy is introduced to reduce possible errors in cluster assignment. Another work based on fuzzy weighted K-nearest neighbour (FKNN-DPC) proposed by Xie et al. [21], focused on the problem of using different measures for local density computation in DPC. They proposed a uniform density metric based on KNN. Du et al. [22] proposed DPC-KNN, which initially introduced the idea of KNN to DPC and provided another option to compute local density. For handling high-dimensional datasets, principal component analysis (PCA) was used during preprocessing for dimensionality reduction. Recently, Hou et al. [23] proposed a density peak clustering algorithm which identifies the cluster centres using a relative density relationship approach and computes the local density using KNN. Wang et al [24] proposed McDPC for identifying clusters having multiple density peaks and low density clusters. McDPC improves the performance of DPC, however, relies on many additional parameters. In [25], authors have proposed a fuzzy kernel based on KNN for computing the local density and introduced a different method to select the cluster centers. Another recent work [26] uses mutual nearest neighbor information (based on KNN) to propose a new clustering algorithm based on DPC to detect arbitrary shaped clusters. The major issue with the above works is that they change the original semantics of DPC.

There exist other variants of DPC that focus on improving its speed on large datasets by reducing the distance computations [27,28], however, they are out of the scope of the study of this paper.

3.3. Limitations of DPC

The first limitation of DPC is the fact that it cannot identify clusters with highly varied densities. Indeed, because DPC selects cluster centers based on high ρ and large δ (or large γ), a cluster in a region in which none of the objects have sufficiently large γ may not be properly identified. Therefore, each object of this cluster will belong to the cluster of its nearest neighbor of higher density. Consequently, either some of the objects of this cluster or the entire cluster may become associated with another cluster. This issue is illustrated in Fig. 1(a) and (b), where in both examples, two cluster centers are incorrectly assigned to the same true cluster, leading to misclassification of many objects.

The second issue of DPC is that its clustering results are sensitive to the change in the units/scales used to measure/represent the features. Indeed, DPC requires the computation of the pairwise distance of objects for finding clusters. Hence, given that the distance between two objects can vary when the units/scales used to measure/represent the feature(s) changes, the clustering results can be significantly different from the results that can be obtained using the data in the original scale. Since the data of real-world applications are obtained from different sources, the units/scales of the features used may be different. For example, feature values can be represented in either log or *inverse* scale (e.g., likelihood can be given as log likelihood, fuel efficiency of vehicles in KM/L or L/100KM). In such cases, it is natural to expect that DPC clustering results will not be affected by the change in units/scales.

While linearly scaled features are easily handled when the data is normalised to a unit scale, the non-linearly scaled features may still be problematic and difficult to resolve. For example, if a, b, c and d are real values in ascending order with equal interval, and a', b', c' and d' are their corresponding values in the logarithmic scale, then we have $b-a=d-c$ but $b'-a' \neq d'-c'$. Consequently, the similarity between two objects captured in different scales can be different, which can significantly affect the clustering results of DPC. The effect of data transformation on the results of DPC is illustrated in Fig. 1(c) and (d). We can observe that the clustering results obtained by DPC on these datasets are significantly different from those obtained on the datasets with the original scales.

One reason for the above-mentioned issues with DPC is because it uses the Euclidean distance for computing the similarity between two objects. A recent study by Aryal et al. [13] has mentioned two issues of using distance as a measure of (dis)similarity between objects in data mining: (i) the similarity of two objects is data distribution independent (i.e., it is not affected by the distribution of other data objects), and (ii) the similarity of two objects is sensitive to the data representation (i.e., units/scales used to measure data features).

Previous work has tried to improve the performance of DPC by incorporating diverse similarity measures [14,21], but most of these methods remain sensitive to units/scales of data. For example, SNNDPC [14] attempts to alleviate the issue of varying density in DPC to some extent by replacing the Euclidean distance-based similarity measure with a new measure based on Shared Nearest Neighbours (SNNs). However, it remains sensitive to the change in units/scales used to express data because SNNs are searched using the Euclidean distance. In addition, SNNDPC modifies some procedures in the original DPC algorithm, which causes the loss of the true semantic of DPC. This necessitates the development/usage of data-dependent similarity measures for DPC that capture the intrinsic structure of the data, thus reflecting the true similarity between objects and helping to achieve better clustering results.

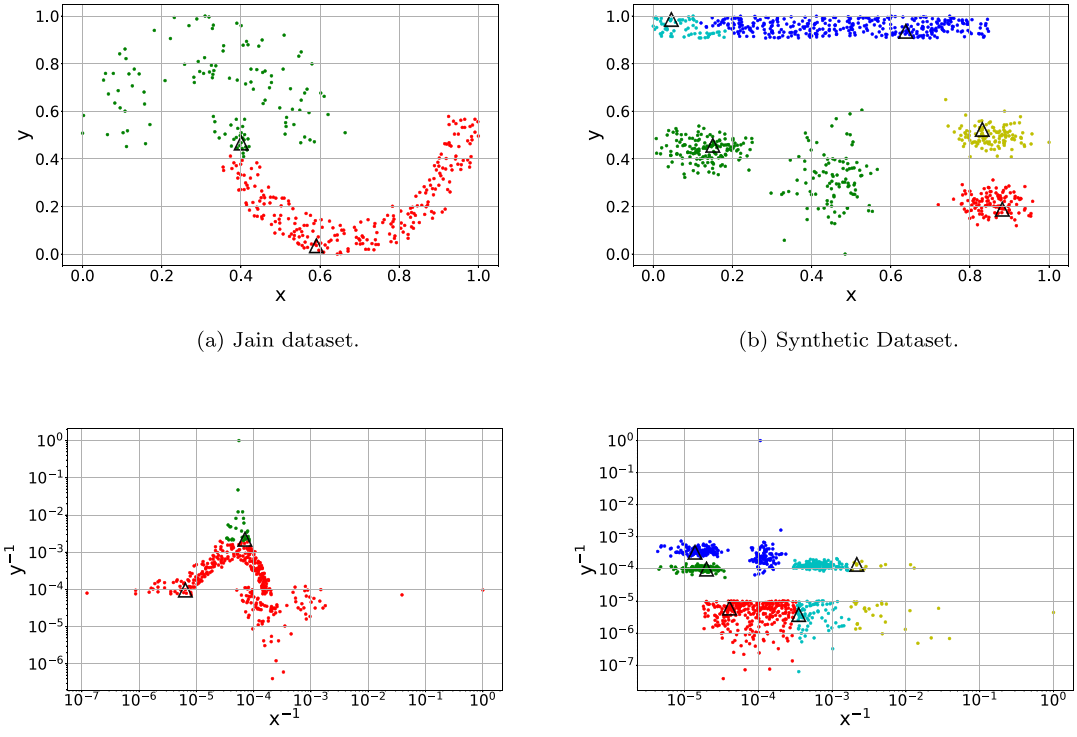
4. MP-DPC

In this section, we propose a data-dependent similarity measure, which we call MP-Similarity. Then, using MP-Similarity we present MP-DPC, which is a data-dependent variant of DPC.

4.1. The idea of data-dependent similarity

We have previously argued that the use of distance-based Euclidean similarity measures is not appropriate for estimating similarities between objects. The similarity estimated by a data-independent measure (such as the Euclidean distance) between two objects $\mathbf{x}$ and $\mathbf{y}$ in a dense region is the same as that of two equidistant objects in a sparse region. However, psychologists argue that the human-judged similarity between two objects is data-dependent [29,30]. In other words, two objects in a sparse region should be more similar than two objects with equal distance in a dense region. For example, consider two groups of dogs such that the first group has only one breed of dogs (e.g., German Shepherd), while the second group has a mix of breeds. A human would judge two dogs in the first group to be less similar (as all dogs are of the same breed) than the two German Shepherd dogs in the second group. Thus, the similarity measure must take into account the distribution of data.

The similarity between two objects in a multidimensional space is estimated by aggregating their similarities in each dimension. If two objects $\mathbf{x}$ and $\mathbf{y}$ have the same value in a dimension i (i.e., $x_i = y_i$), a data-independent similarity measure assigns the similarity of 1 regardless of the distribution of data in the dimension i. However, as mentioned earlier, it should depend on the data distribution (i.e., the likelihood of x_i). For example, consider the data distribution shown in Table 1, where the objects are represented in the columns and the values of the features/dimensions in the rows. In particular, let us consider the two instances Inst1 and Inst2, that



(a) Jain dataset. (b) Synthetic Dataset.

(c) Jain dataset with scaled features. (d) Synthetic dataset with scaled features.

Fig. 1. Clustering results of the traditional DPC algorithm [First Row] Jain and Synthetic datasets with diverse densities shown in Subfigures (a) and (b) and [Second Row] Jain and Synthetic datasets with features transformed to inverse scale shown in Subfigures (c) and (d). Note that x^{-1} and y^{-1} are plotted on the logarithmic scale. Each Δ represents a cluster center.

Table 1
Example of a sample data distribution to show the importance of data-dependent similarity measure [13].

Dim.	*Inst1*	*Inst2*	*Inst3*	*Inst4*	*Inst5*	*Inst6*	*Inst7*	*Inst8*	*Inst9*	*Inst10*
.	.	.	.	.	.	.	.	.	.	.
i	2	2	1	1	1	1	1	1	1	1
j	2	2	2	2	2	2	2	2	1	1
.	.	.	.	.	.	.	.	.	.	.

take the same values in both the ith and jth dimension, however their matching value in the ith dimension is rare (only these two instances have the value of 2) and that in the jth dimension is frequent (these two and six other instances have the value of 2). When estimating the (dis)similarity between *Inst1* and *Inst2* using the Euclidean distance (data-independent measure), the matching values in dimensions i and j contribute equally to the overall similarity score. However, psychologists argue that they do not provide the same amount of information about the similarity of *Inst1* and *Inst2* and they should contribute differently. The matching value in dimension i is rare (low probability) and it provides more information than the matching value in dimension j which is frequent (high probability). This is especially true in the case of high-dimensional datasets in which the objects mostly lie in low-dimensional manifolds. Therefore, from the above discussion, similarity measures that take into account the data distribution both when $x_i \neq y_i$ and $x_i = y_i$ are useful to represent the intrinsic structure of the data.

4.2. Mass-based probabilistic (MP) similarity measure

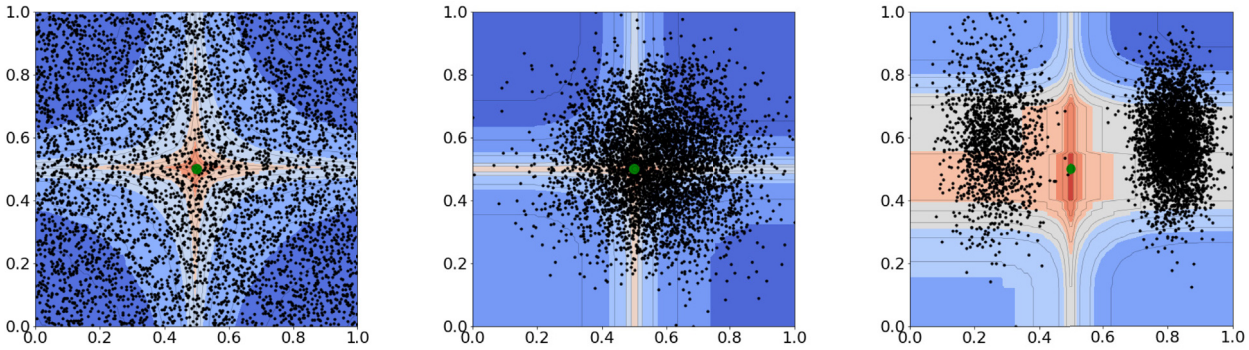
In this section, we first discuss the data-dependent dissimilarity measure known as m_0-dissimilarity, and then, we introduce our Mass-based Probabilistic Similarity (MP-Similarity) measure.

4.2.1. m_0-Dissimilarity

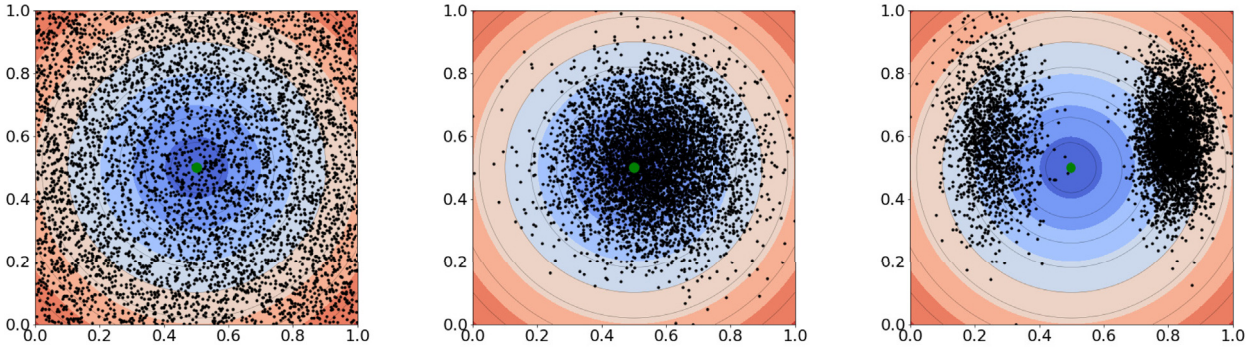
Aryal et al. [13] proposed m_0-dissimilarity as a fully data-dependent dissimilarity measure and used it in the context of KNN classification. It estimates the dissimilarity of two objects $\mathbf{x}$ and $\mathbf{y}$ by using the probability mass of the region covering x_i and y_i in each dimension i instead of the spatial distance $|x_i - y_i|$. These dissimilarities are then aggregated to give the m_0 dissimilarity. The idea is that $\mathbf{x}$ and $\mathbf{y}$ are more dissimilar with respect to the dimension i if many objects in the dataset have the values of feature i between x_i and y_i. Let m be the number of objects in the dataset $\mathcal{D}$, n be the number of dimensions, $R_i(\mathbf{x},\mathbf{y}) = [\min(x_i, y_i), \max(x_i, y_i)]$ be the region covering objects $\mathbf{x}$ and $\mathbf{y}$ in dimension i and $|R_i(\mathbf{x}, \mathbf{y})| = |\{\mathbf{z} \in P : \min(x_i, y_i) \leq z_i \leq \max(x_i, y_i)\}|$, then m_0-dissimilarity of objects $\mathbf{x}$ and $\mathbf{y}$ is defined as:

$$m_0(\mathbf{x},\mathbf{y}) = \left(\frac{1}{n} \sum_{i=1}^{n} \log \frac{|R_i(\mathbf{x}, \mathbf{y})|}{m} \right) \tag{4}$$

In the above equation, $\frac{|R_i(\mathbf{x}, \mathbf{y})|}{m}$ represents the probability mass of the region in which x_i and y_i lie. Aryal et al. [13] provided a probabilistic interpretation of m_0-dissimilarity based on the Naive Bayesian assumption that attributes (dimensions) are independent of each other.



(a) Uniform distribution (MP). (b) Normal distribution (MP). (c) Varying density Clusters (MP).



(d) Uniform distribution (Euclidean). (e) Normal distribution (Euclidean). (f) Varying density Clusters (Euclidean).

Fig. 2. Contour plots of dissimilarity of objects in the 2-dimensional space with reference to object (0.5, 0.5), based on MP similarity (first row) and Euclidean distance (second row), in different distributions. The darker the color, the higher the dissimilarity.

In the above equation, $|R_i(\mathbf{x}, \mathbf{y})|$ incorporates the region information in which x_i and y_i lie for each dimension i, which makes m_0 a data-dependent measure. In the example shown in Table 1, the same values in dimensions i and j contribute differently in the overall dissimilarity of *Inst1* and *Inst2*. Moreover, m_0 is not affected when the values x_is along feature i are scaled (linearly or nonlinearly) to x_i's (a different scale) as the number of instances in $|R_i(\mathbf{x}, \mathbf{y})|$ and $|R_i(\mathbf{x}', \mathbf{y}')|$ remains the same because scaling either preserves or reverses the ordering of data values. Thus, m_0 captures the data-distribution as well as remain invariant to any change in the scale of data.

Computing $|R_i(\mathbf{x}, \mathbf{y})|$ requires a range search which makes it computationally expensive. To efficiently compute $|R_i(\mathbf{x}, \mathbf{y})|$, Aryal et al. [13] proposed to divide the range of continuous valued domain in the ith dimension into b equal-frequency intervals or bins. For each bin, the proposed algorithm stores the frequency or the number of objects in that bin. Equal-frequency binning makes it invariant to linear or non-linear scaling of data. Because many data objects can have the same value in feature i (i.e., there can be duplicate values), it may be impossible to have perfectly equal-frequency bins, i.e., bins may not always have the same frequency. With bin frequencies, $|R_i(\mathbf{x}, \mathbf{y})|$ can be approximated quickly by aggregating the frequencies of the bins in which x_i and y_i lie and the bins in between. Data mass between each pair of bins in dimension i can be precomputed and stored in a $b \times b$ matrix so that $|R_i(\mathbf{x}, \mathbf{y})|$ can be calculated directly by a matrix lookup.

4.2.2. MP-Similarity

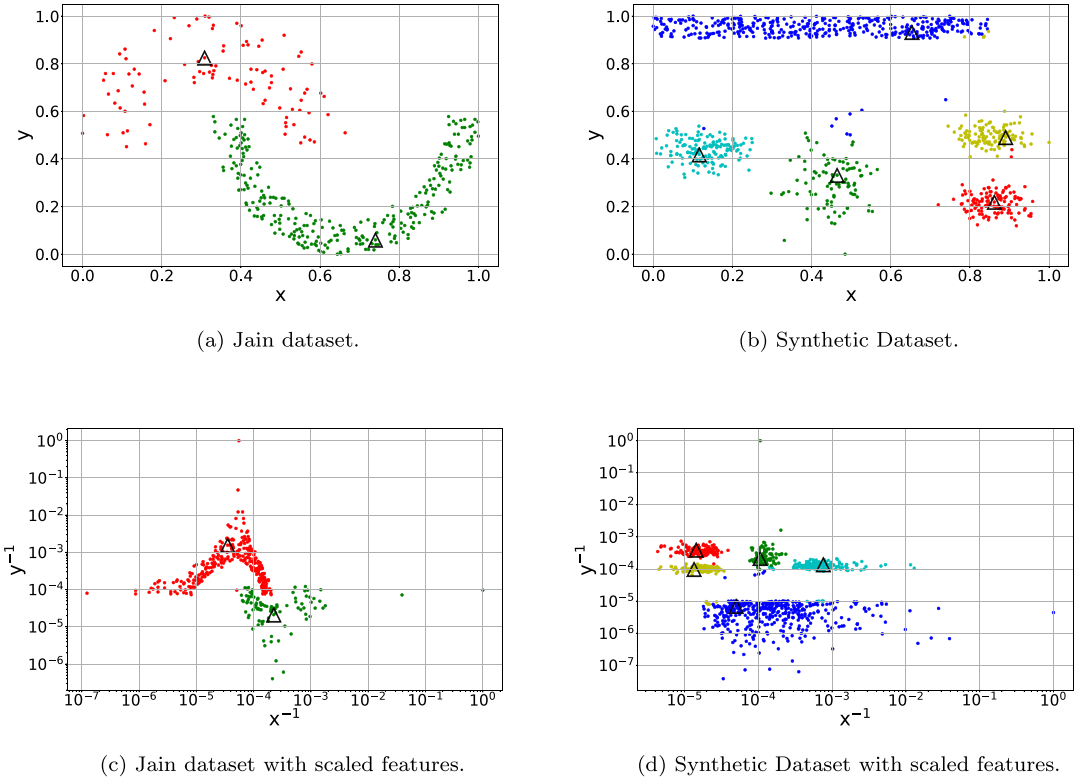
We observe that m_0 is not a valid metric as: (i) the dissimilarity $m_0(\mathbf{x},\mathbf{y})$ of objects when $\mathbf{x} = \mathbf{y}$ is non-zero, and (ii) the dissimilarities $m_0(\mathbf{x},\mathbf{x})$ and $m_0(\mathbf{y},\mathbf{y})$ may not be similar. Therefore, m_0 cannot be used as a similarity measure in applications where metric properties are required/assumed. Motivated by this, we propose a new similarity measure called Mass-based Probabilistic Similarity (MP-Similarity or MP), which is obtained by normalising m_0 using "$m_0(\mathbf{x},\mathbf{x}) + m_0(\mathbf{y},\mathbf{y})$" where $m_0(\mathbf{x},\mathbf{x})$ represents the self-dissimilarity. It is defined as:

$$MP(\mathbf{x}, \mathbf{y}) = \frac{2 * m_0(\mathbf{x},\mathbf{y})}{m_0(\mathbf{x},\mathbf{x}) + m_0(\mathbf{y},\mathbf{y})} \tag{5}$$

Thus, MP retains the merits of m_0, and like m_0-dissimilarity, MP satisfies the symmetric and triangle inequality assumptions – given x_i, y_i, and z_i in dimension i, $|R(x_i, y_i)| = |R(y_i, x_i)|$ and $|R(x_i, z_i)| \leq |R(x_i, y_i)| + |R(y_i, z_i)|$, which can be generalised in a multidimensional space. Also, from Eq. (5), we have that $\forall \mathbf{x}, MP(\mathbf{x},\mathbf{x}) = 1$, which shows that MP satisfies the identity assumption. The range of similarity assigned by MP lies in the range of [0,1] with $MP(\mathbf{x}, \mathbf{x}) = MP(\mathbf{y}, \mathbf{y}) = 1$.

Next, we analyze the behaviour of MP-Similarity under different data distributions including varying density clusters. To do so, we generate three datasets of 5000 instances with 2 dimensions each from the following distributions: (i) uniform distribution, (ii) normal distribution, and (iii) varying density clusters. Figure 2 shows the contour plots of similarity of objects in the space to the center (0.5, 0.5) using MP-Similarity (Fig. 2(a)–(c)) and the Euclidean distance (Fig. 2(d)–(f)). In the contour plots, we represent the region of high similarity with dark orange color and the region of low similarity with dark blue color. From Fig. 2, we make the following observations:

- **MP-Similarity adapts to different data distributions.** Figures 2 and (b) show the contours of MP-Similarity on datasets with uniform distribution and normal distribution re-



(a) Jain dataset. (b) Synthetic Dataset.

(c) Jain dataset with scaled features. (d) Synthetic Dataset with scaled features.

Fig. 3. Clustering results of the MP-DPC algorithm on (i) Jain and Synthetic datasets with diverse densities shown in Figures (a) and (b) and (ii) Jain and Synthetic datasets with features transformed to inverse scale. Note that x^{-1} and y^{-1} are plotted on the logarithmic scale. Each Δ represents a cluster center.

spectively. As it can be observed, the contours are different for the different data distributions which demonstrates the data-dependent behaviour of MP-Similarity as it adapts the contours to underlying data distribution. However, the data-independent similarity measures (e.g., Euclidean distance) produce the same contours irrespective of the different distributions.

- **MP-Similarity adapts to different densities.** Figure 2(c) shows dataset containing two clusters with varied densities. As shown, from the center, the contour decreases slower in the region with sparse concentration of objects than in the region with dense concentration. Thus, MP-Similarity adapts its contours to the local data distribution which is not the case with a data-independent similarity measure, where contour vary at the same rate on either side regardless of the varying data concentration. This reflects the characteristic of a data-dependent measure discussed earlier: two objects in a sparse region are more similar than two objects with equal distance in a dense region.

The data-dependent characteristic of MP-Similarity makes it a suitable candidate for finding similarity between objects regardless of the underlying data distribution. Next, we show that MP-Similarity can be incorporated with DPC to improve its clustering results on datasets with varying density and scales.

4.3. MP-DPC

In this section, we present MP-DPC, which incorporates the MP-Similarity measure in DPC to create a data-dependent version of DPC.

DPC requires pairwise distances/dissimilarities between objects to compute ρ and δ as given in Eqs. (1) and (2). To do so, DPC uses the default Euclidean distance as the distance measure. We replace the default distance measure in DPC by MP-Similarity and we use it as follows:

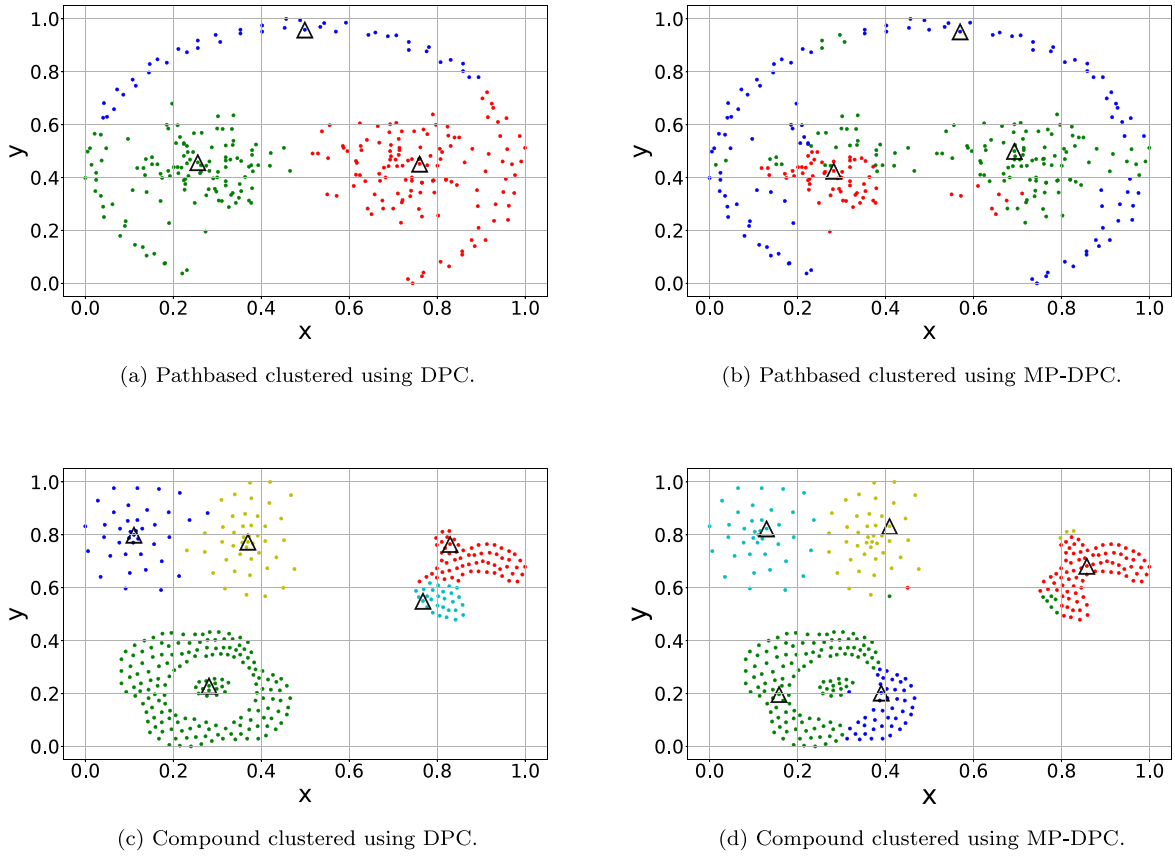
$$dist(\mathbf{x},\mathbf{y}) = 1 - \text{MP}(\mathbf{x},\mathbf{y}) \tag{6}$$

This version of DPC is called MP-DPC, and we argue that it does not require modifying the local density computation (or any other steps) in the procedure of the original DPC algorithm.

Figure 3 shows the performance of MP-DPC on the Jain dataset and a Synthetic dataset that have various densities. Unlike DPC, we observe in Fig. 3(a) and (b) that MP-DPC is clearly able to identify the dense and sparse clusters in both datasets as it is able to adapt to the local data distribution. Similarly, when the features of these datasets are converted to inverse scale as shown in Fig. 3(c) and (d), MP-DPC is still able to identify the same clusters as in the original scale of the features. This is because MP-Similarity is dependent on the number of instances falling between two objects which do not vary when the scale is varied linearly or non-linearly. Thus, MP-DPC is suitable for varying data distributions, densities and scale.

Next, we show in Fig. 4 the performance of MP-DPC on two complex shape datasets with multi-density clusters: the Path-based dataset, which consists of three clusters and the Compound dataset, which consists of five clusters. We observe that on the Pathbased dataset, MP-DPC identifies the ring cluster better than DPC but shows some misclassifications on the other two clusters inside the ring. For the Compound dataset, we observe that both DPC and MP-DPC did not perform well: they both correctly identified the two sparse clusters on the upper left corner, but they both failed to correctly identify the other clusters. This performance of DPC and MP-DPC could be an issue of the method of selection of cluster centers, which we need to further investigate.

We also incorporate MP-Similarity in SNNDPC [14] to improve its performance and we call this new version MP-SNNDPC. The remaining steps of SNNDPC do not need to be modified as in MP-DPC.



(a) Pathbased clustered using DPC.

(b) Pathbased clustered using MP-DPC.

(c) Compound clustered using DPC.

(d) Compound clustered using MP-DPC.

Fig. 4. Performance of DPC and MP-DPC algorithm on multi-density datasets Pathbased and Compound.

Table 2
Summary of the Datasets.

Dataset	#Instances	#Attributes	#Clusters
Iris	150	4	3
Wine	178	11	3
Parkinson	197	23	2
Thyroid	215	4	3
Libra	360	90	15
Dermatology	366	33	6
WDBC	569	30	2
Balance Scale	625	4	3
Statlog (Vehicle)	846	18	4
Gtzan	1000	230	10
Hba	1500	187	15
Wap	1560	8460	20
Cardiotocography	2126	23	10
Fbis	2463	2000	17
Spambase	4601	58	2
Satimage	6435	36	6
Corel	10,000	67	100

Finally, we note that the time complexity of MP-DPC and MP-SNNDPC remains the same as their original counterparts based on the Euclidean distance because $m_0(\mathbf{x},\mathbf{y})$ can be estimated in $O(n)$ time.

5. Experimental evaluation

5.1. Experimental setup

In this section, we describe the experimental setup we used in our evaluations, including a description of the baselines, details of our implementation, the datasets, and the metrics used.

5.1.1. Baselines

We demonstrate the performance of our MP-DPC and MP-SNNDPC algorithms by comparing them against the following state-of-the-art algorithms:

- DPC: the original DPC algorithm using the Euclidean distance;
- SNNDPC: the original SNNDPC algorithm using the Euclidean distance;
- IK-DPC: DPC algorithm using a similarity measure based on Isolation Kernel (IK) [31];
- IK-SNNDPC: SNNDPC algorithm using the IK measure;
- Lin-DPC: based on Lin's similarity measure [32];
- Lin-SNNDPC: SNNDPC using Lin's similarity measure;
- k-means [16]: which is used as a simple baseline.

We note that there exist other data-dependent measures, which can be categorized into one-dimensional and tree-based measures [13]. We have selected IK as it is a popular tree-based measure [31] and Lin as it is based on a similar idea as MP.

Specifically, IK [31] is a data-dependent similarity measure based on an ensemble of t random trees called Isolation Forest [33]. Each tree $H_i(i = 1, 2, \ldots t)$ partitions the space using a small subsample of data $\mathcal{D}_i \subset \mathcal{D}$ ($|\mathcal{D}_i| = \psi$). Partitions are created such that sparse regions comprise of larger partitions than the dense regions adapting to local data distribution. The similarity of two objects is estimated as the number of trees in which the two objects fall in the same partition.

Regarding the Lin similarity measure [32], it introduces a notion of similarity based on information theory for ordinal data. Aryal et al. [13] presented its multidimensional version to measure the similarity of two data objects $\mathbf{x}$ and $\mathbf{y}$ in multidimensional continuous domain, which aggregates the Lin's similarity in each dimen-

Table 3
AMI: Performance of Clustering algorithms and their data-dependent variants.

Datasets	*k*-means	DPC				*k*-means	SNNDPC			
		Euclidean	IK	Lin	MP		Euclidean	IK	Lin	MP
Iris	0.7387	0.7810	0.8286	**0.8479**	**0.8479**	0.7387	**0.9133**	0.7696	0.8968	0.8625
Wine	**0.8514**	0.7847	0.7702	0.6752	0.7070	0.8514	0.8769	0.8101	**0.9209**	0.9099
Parkinson	0.2318	0.1916	**0.2833**	0.2769	0.2115	0.2318	0.2637	0.2193	**0.2873**	0.2821
Thyroid	0.5909	0.3097	0.7530	0.7858	**0.7947**	0.5909	0.5858	0.6318	**0.8497**	0.8076
Libra	0.5399	0.5706	0.5234	0.5210	**0.5264**	0.5399	**0.6283**	0.5407	0.5237	0.5299
Dermatology	0.8811	0.8096	0.8121	0.8820	**0.9370**	0.8811	0.8990	0.8509	0.9108	**0.9301**
WDBC	0.6226	0.2447	0.6342	**0.7182**	0.6614	0.6226	**0.7335**	0.5872	0.6983	0.6933
Balance Scale	0.1116	0.1763	0.1002	**0.2144**	**0.2144**	0.1116	0.1726	0.1502	**0.1876**	0.1809
Statlog	0.0966	0.1735	0.2129	0.3368	**0.3413**	0.0966	0.2050	0.2457	**0.3405**	0.3230
Gtzan	**0.3288**	0.2738	0.1667	0.2785	0.2813	**0.3288**	0.3179	0.1969	0.2965	0.2944
Hba	0.3018	0.1977	0.2280	0.3024	**0.3097**	0.3018	0.2424	0.2743	**0.3653**	0.3556
Wap	**0.3143**	0.1147	0.1269	0.1415	0.2908	0.3143	0.1203	0.1383	0.2119	**0.3798**
Cardiotocography	0.2763	0.2241	0.2764	0.2940	**0.3200**	0.2763	0.2391	0.3157	**0.3571**	0.3510
Fbis	0.1762	0.3163	0.1189	0.2589	**0.3442**	0.1762	**0.2319**	0.0222	0.0280	0.2095
Spambase	0.0098	0.0950	0.0758	0.2186	**0.3815**	0.0098	0.0737	0.1568	0.2065	**0.3633**
Satimage	0.6119	**0.6377**	0.6194	0.5891	0.5958	0.6119	0.6475	0.5807	0.6816	**0.7107**
Corel	0.1973	0.1473	0.1753	0.2018	**0.2079**	0.1973	0.1743	0.2162	0.2493	**0.2514**
Avg.	0.4048	0.3558	0.3944	0.4437	0.4690	0.4048	0.4309	0.3945	0.4713	0.4962

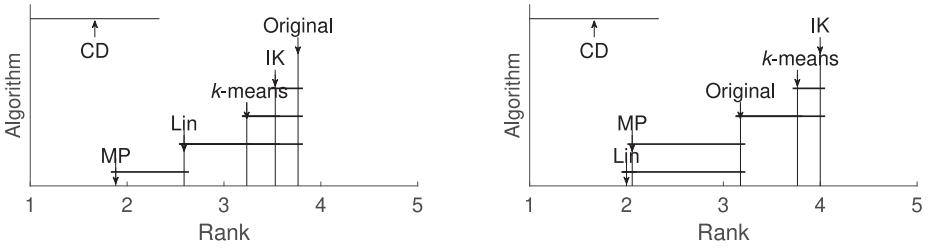


Fig. 5. Post-hoc Nemenyi test ($\alpha = 0.10$) based on AMI scores shown in Table 4.

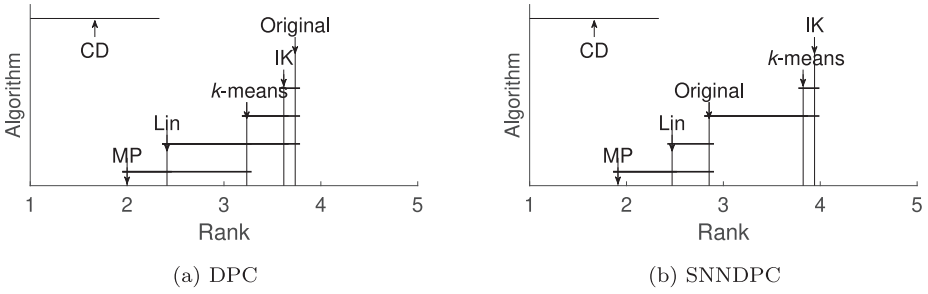


Fig. 6. Post-hoc Nemenyi test ($\alpha = 0.10$) based on ARI scores in Table 4.

sion as shown below:

$$Lin(\mathbf{x},\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n}\frac{2\times\log\sum_{z_i=\min(x_i,y_i)}^{\max(x_i,y_i)}P(z_i)}{\log P(x_i)+\log P(y_i)} \tag{7}$$

We use the idea of equal-frequency bins and data mass in bins as in the case of MP-Similarity to compute $P(z_i)$, $P(x_i)$ and $P(y_i)$ efficiently. The fundamental difference between these two measures is the ability of MP-Similarity to capture data-dependent information in features where $x_i = y_i$. In Lin's measure, the similarity of $\mathbf{x}$ and $\mathbf{y}$ in a dimension i is 1 regardless of the likelihood of x_i. In the example discussed in Table 1, both features i and j contribute equally to the overall similarity of $Inst1$ and $Inst2$ though they share a frequent value in the dimension j and a rare value in the dimension i. But, they contribute differently in MP-Similarity.

5.1.2. Implementation details

The implementations of SNNDPC and *k*-means we used are based on [14] and [34] respectively. The implementation of IK is based on the tree-partition mechanism of Isolation Forest as described in [31]. The implementation of Lin is similar to the implementation of MP as both require $|R_i(\mathbf{x},\mathbf{y})|$ for the computation of similarity of two objects x and y in dimension i.

All algorithms requires parameters tuning to obtain optimal performance. Specifically, DPC requires the parameter d_c to be prespecified. Authors of the original DPC paper [5] suggested d_c to be such that the total number of neighbours are between 1–2% of the total number of points. Extensions of DPC [14,21,23] have modified the percentage to obtain the best results. We modify and extend this range from 1% to 3% with step size of 0.1% and report the best results. SNNDPC requires tuning the parameter k for obtaining the best results [14], which we selected in the range of [5, 50] with a step size of 5. For consistency, we use the same setting of d_c and k with the data-dependent and data-independent variants of DPC and SNNDPC.

For MP-DPC and Lin-DPC, MP-Similarity and Lin's measure require setting an appropriate number of bins b. We vary b from

Table 4
ARI: Performance of Clustering algorithms and their data-dependent variants.

Datasets	*k*-means	DPC				*k*-means	SNNDPC			
		Euclidean	IK	Lin	MP		Euclidean	IK	Lin	MP
Iris	0.7163	0.7196	0.8266	**0.8681**	**0.8681**	0.7163	**0.9222**	0.7498	**0.9222**	0.8857
Wine	**0.8685**	0.8191	0.7627	0.6661	0.6999	0.8685	0.8922	0.8226	0.9284	**0.9295**
Parkinson	0.0520	0.3363	0.3758	**0.3811**	0.2879	0.0520	0.2916	0.2172	0.2148	**0.2985**
Thyroid	0.6283	0.1822	0.8256	0.8601	**0.8623**	0.6283	0.6823	0.6638	**0.9064**	0.8618
Libra	0.3207	**0.3503**	0.3055	0.3024	0.3018	0.3207	**0.4134**	0.3141	0.3012	0.2914
Dermatology	0.7426	0.8159	0.7912	0.8785	**0.9409**	0.7426	0.8434	0.8216	0.8637	**0.9302**
WDBC	0.7302	0.2712	0.7327	**0.8183**	0.7741	0.7302	**0.8308**	0.6724	0.8052	0.8055
Balance Scale	0.1351	0.1348	0.1138	**0.2934**	**0.2934**	0.1351	**0.2203**	0.1885	0.1871	0.1997
Statlog	0.0757	0.1092	0.1531	0.2286	**0.2346**	0.0757	0.1509	0.1762	**0.2813**	0.2653
Gtzan	**0.1876**	0.1580	0.0915	0.1700	0.1670	**0.1876**	0.1513	0.0673	0.1391	0.1391
Hba	**0.1471**	0.1087	0.1087	0.1396	0.1454	0.1471	0.079	0.0897	**0.2016**	0.1828
Wap	0.1485	0.0555	0.0699	0.0638	**0.2368**	0.1485	0.0551	0.0479	0.1447	**0.1831**
Cardiotocography	0.1317	0.0820	0.1320	0.1651	**0.1889**	0.1317	0.1248	0.1622	**0.2315**	0.1964
Fbis	0.0493	0.1227	0.0401	0.129	**0.2437**	0.0493	**0.1111**	-0.0014	-0.0031	0.0942
Spambase	-0.0048	0.1514	0.0654	0.2987	**0.4344**	-0.0048	-0.003	0.1297	0.2427	**0.4492**
Satimage	**0.5263**	0.5120	0.5139	0.5016	0.5198	0.5263	0.5824	0.4937	0.6660	**0.7021**
Corel	0.0462	0.0307	0.0297	**0.0493**	0.0489	0.0462	0.0253	0.0361	0.0553	**0.0619**
Avg	0.3236	0.2917	0.3493	0.4008	0.4263	0.3236	0.3749	0.3324	0.4169	0.4398

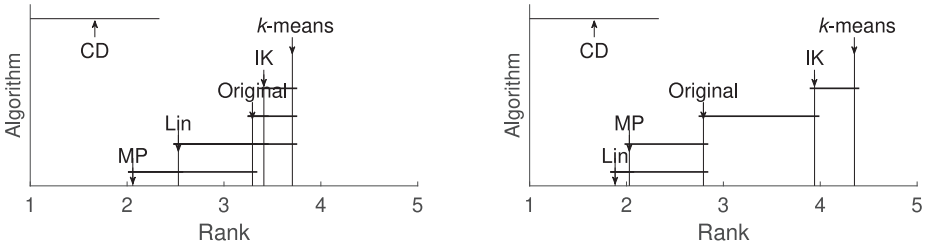


Fig. 7. Post-hoc Nemenyi test ($\alpha = 0.10$) based on FMI scores in Table 5.

the set [20, 40, 60, 80, 100, $log_2(m)$], where $log_2(m)$ is the default value used in [13]. IK has two parameters, the *sampling size* (ψ) and the *number of trees* (t). For ψ, we select a value from $\{2^a | a = 2, 3, \ldots, 8\}$, while the number of trees is fixed to the default value of 100, as suggested in [31,33]. Since the algorithm is random, we run it 5 times for each sampling size and report the mean score for each evaluation metric discussed in Section 5.1.4.

In *k*-means, the initial centres are selected using the *k*-means++ [35] method. Additionally, all algorithms require specifying the number of cluster centers. To be consistent, we fixed the number of clusters in these algorithms to be equal to the number of classes in the ground truth.

5.1.3. Datasets

The above algorithms are evaluated on 17 real-world datasets, which are described in Table 2. Many of these datasets are widely used in existing DPC literature, such as [14,21,23], and are obtained from UCI Machine Learning Repository [36]. These datasets are different in terms of number of objects, dimensionality, and the number of clusters. The varying properties of these datasets help to

Table 5
FMI: DPC algorithm with data-independent Euclidean distance and data-dependent (dis)similarity measures.

Datasets	*k*-means	DPC				*k*-means	SNNDPC			
		Euclidean	IK	Lin	MP		Euclidean	IK	Lin	MP
Iris	0.8112	0.8159	0.8846	**0.9115**	**0.9115**	0.8112	**0.9479**	0.8350	0.9478	0.9233
Wine	**0.9126**	0.8801	0.8452	0.7800	0.7952	0.9126	0.9330	0.8825	0.9525	**0.9532**
Parkinson	0.5957	**0.8140**	0.8059	0.7433	0.6982	0.5957	**0.8167**	0.6932	0.7584	0.7036
Thyroid	0.8546	0.5697	0.9206	**0.9356**	0.9351	0.8546	0.8611	0.8525	**0.9572**	0.9349
Libra	0.3726	**0.4009**	0.3627	0.3564	0.3547	0.3726	**0.4598**	0.3649	0.3605	0.3505
Dermatology	0.7947	0.8519	0.8325	0.9034	**0.9526**	0.7947	0.8824	0.8572	0.8944	**0.9441**
WDBC	0.8770	0.6595	0.8778	**0.9142**	0.8938	0.8770	**0.9217**	0.8467	0.9084	0.9085
Balance Scale	0.4666	0.5478	0.5042	**0.6025**	**0.6025**	0.4666	0.5251	0.5105	**0.6552**	0.6533
Statlog	0.3070	0.4209	0.3872	0.4540	**0.4610**	0.3070	0.3881	0.4281	**0.4793**	0.4713
Gtzan	0.2942	0.2971	0.2326	0.3020	**0.3028**	0.2942	0.3226	0.3005	**0.3343**	**0.3343**
Hba	0.2351	0.2015	0.2084	0.2350	**0.2410**	0.2351	0.2359	0.2086	**0.2772**	0.2713
Wap	**0.3642**	0.1961	0.2706	0.2615	0.3410	0.3642	0.3122	0.3056	0.3122	**0.3747**
Cardiotocography	0.2498	0.2310	0.2773	0.3265	**0.3335**	0.2498	0.2531	0.2814	**0.3689**	0.3233
Fbis	0.1916	0.2779	0.2588	0.2693	**0.3309**	0.1916	0.2917	0.3176	**0.3239**	0.2886
Spambase	0.7160	0.7179	0.6931	0.7221	**0.7537**	0.7160	0.7186	0.7213	0.7224	**0.7551**
Satimage	0.6142	**0.6299**	0.6145	0.6032	0.6095	0.6142	0.6934	0.5986	0.7382	**0.7677**
Corel	0.0564	0.0514	0.0622	**0.0739**	0.0728	0.0564	0.0791	0.0803	0.0929	**0.0939**
Avg	0.5126	0.5037	0.5317	0.5526	0.5641	0.5126	0.5672	0.5344	0.5932	0.5913

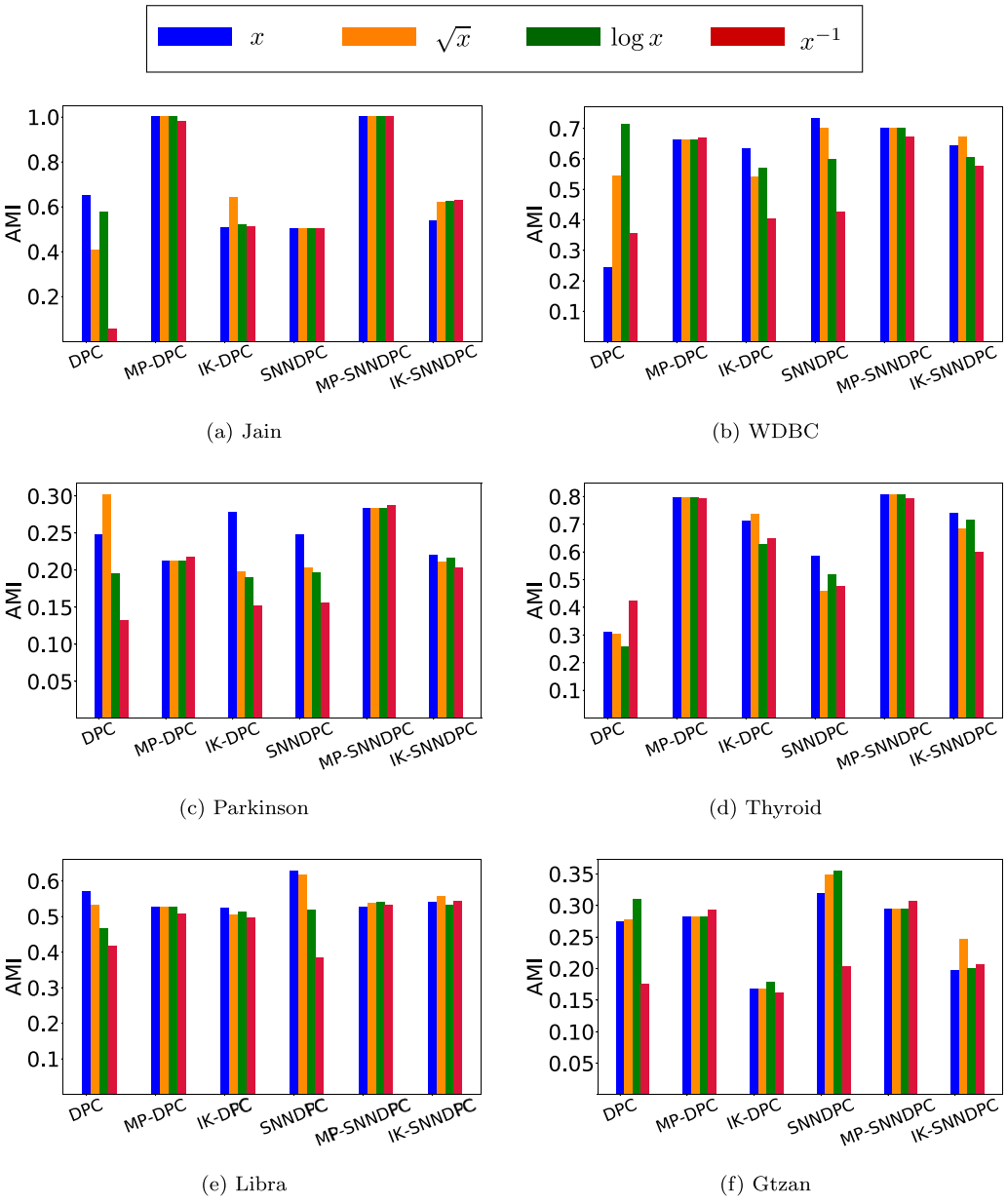


Fig. 8. Variation in AMI score of algorithms after scaling features as $\sqrt{x}$, $\log x$ and x^{-1}.

comprehensively evaluate the performance of the clustering algorithms, as they can represent different situations.

To eliminate large differences in the range of dimensions, we normalise the feature values of the datasets to be in the range of [0,1] using min-max normalisation. Note that normalisation is not required for MP-Similarity as it does not use the data values in the similarity calculation, but it is important for distance-based measures such as Euclidean distance.

5.1.4. Evaluation metrics

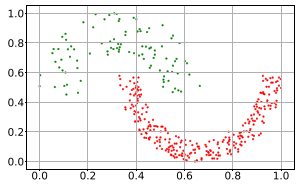
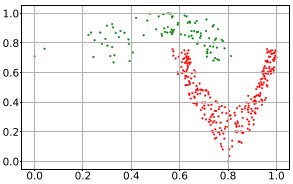
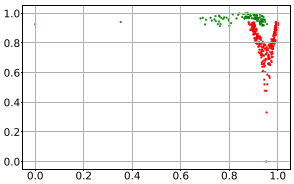
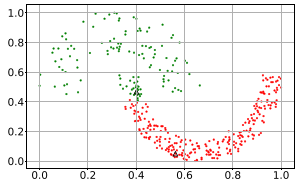
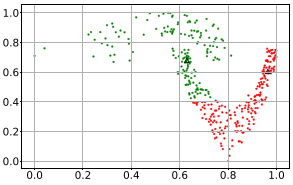
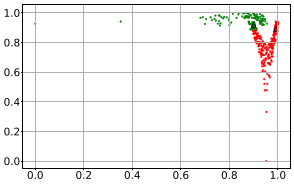
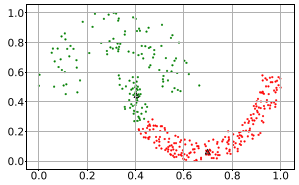
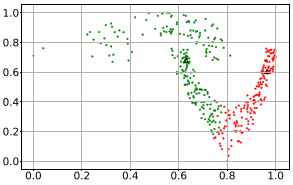
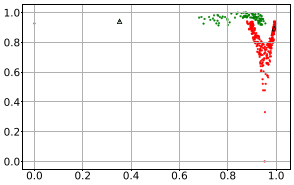
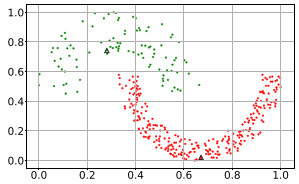
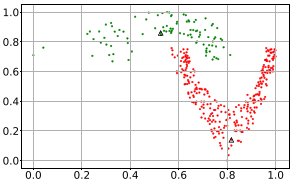
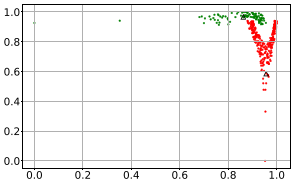
We evaluate the performance of all clustering algorithms using the following popular metrics: Adjusted Mutual Information, Adjusted Rand Score and Fowlkes-Mallows Index. Specifically, let's U and V be respectively the ground truth cluster labels and cluster assignments by a clustering algorithm for m data objects, a be the number of object pairs that belong to the same cluster in U and V, b be the number of object pairs that belong to the same cluster in U but not in V, c be the number of object pairs that belong to the same cluster in V but not in U, d be the number of object pairs belonging to different clusters in U and V. The metrics are defined as follows:

- Adjusted Mutual Information (AMI): AMI is used to measure the similarity between two clusterings of the same data. It is a variation of Mutual Information (MI) that informs the reduction in the Entropy (H) of class labels when the class labels are known. Unlike mutual information, AMI is adjusted against chance which means that the similarity is only governed by the structure of the dataset appearing in two clustering and not by chance. AMI is calculated as follows [34]:

$$AMI = \frac{MI(U,V) - \mathbb{E}[MI(U,V)]}{\frac{1}{2}(H(U)+H(V)) - \mathbb{E}[MI(U,V)]} \tag{8}$$

  where $\mathbb{E}[MI(U,V)]$ is the expectation of the mutual information.
- Adjusted Rand Index (ARI): ARI [37] is another metric to evaluate the similarity of the clusterings. It considers the number of objects existing in the same cluster and in different clusters,

Table 6
Clustering results of the algorithms before and after feature scaling in the Jain dataset.

	x	$\sqrt{x}$	$\log x$
Ground truth			
DPC			
SNNDPC			
MP-DPC			

and measures the fraction of pairs of points that are correctly clustered to the same or different clusters.

$$ARI = \frac{a - (a+c)(a+b)/d}{(a+c) + (a+b)/2 - (a+c)(a+b)/d} \tag{9}$$

- Fowlkes Mallows Index (FMI): FMI [38] is used to measure the similarity of clusters obtained through different clustering algorithms. It is computed as follows:

$$FMI = \frac{a}{\sqrt{(a+b)(a+c)}} \tag{10}$$

A perfect clustering will result in achieving a maximum score of 1 for the above metrics.

5.2. Results

In this section, we report the results of our experiments comparing the performance of our proposed MP-Similarity measure in DPC and SNNDPC algorithms with other contenders.

5.2.1. Performance analysis

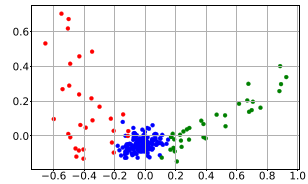
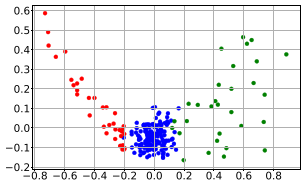
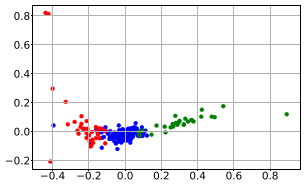
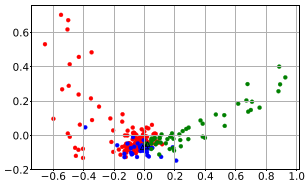
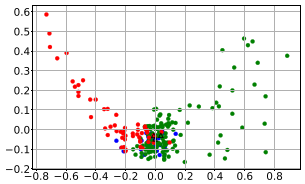
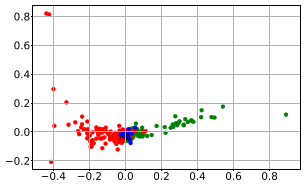
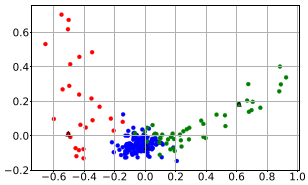
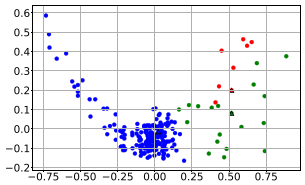
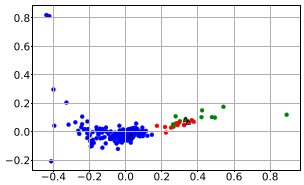
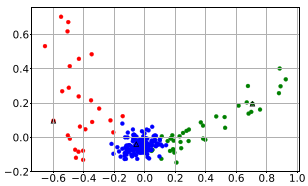
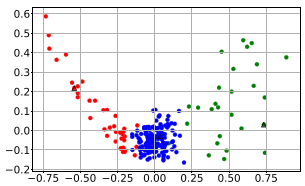
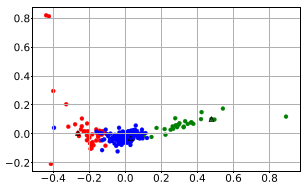
The obtained clustering results are shown in Tables 3, 4, and 5, for respectively AMI, ARI, and FMI. The different variants of DPC and SNNDPC are denoted by the name of the similarity measure they use. The average score over all the datasets is presented in the last row of each table.

Table 3 shows the AMI score obtained by the DPC and the SNNDPC algorithm using the different similarity measures (discussed earlier) and the k-means algorithm on different datasets. The results show that MP-DPC outperforms other variants of DPC and k-means on most datasets. An interesting observation is that all the data-dependent variants have generally better performance than that of the Euclidean. While IK could not achieve the best score on any dataset, it shows better results than the Euclidean on many datasets. k-means obtained best score on three datasets but has less overall average score than the MP and Lin. On the other hand, for SNNDPC, Lin has the best score on seven datasets, MP on five datasets and Euclidean on four datasets. However, MP achieves better average score than any of the contenders including Lin, demonstrating its superior performance.

In terms of ARI (Table 4) and FMI score (Table 5), similar behaviour is observed. While the performance of algorithms varied for some datasets, MP-DPC and MP-SNNDPC still outperform their other counterparts as well as k-means algorithm on most datasets. The close competitor to MP in all the three metrics is Lin as they both consider the probability mass of the objects in each dimension.

Further, we conduct the Friedman test with the post-hoc Nemenyi test [39] to evaluate whether the performance difference between any two algorithms is significant based on the three metrics. For each metric, we show the results of significance test for DPC and SNNDPC clustering algorithms in Figs. 5, 6, and 7. We note that two algorithms are significantly different if there is not

Table 7
Clustering results of the algorithms before and after feature scaling in the Thyroid dataset.

	x	$\sqrt{x}$	$\log x$
Ground truth			
DPC			
SNNDPC			
MP-DPC			

a line linking them. The results show that the MP variants of the algorithms are better than at least two other clustering algorithms.

5.2.2. Sensitivity to varying feature scales

In this section, we compare the performances of the MP variants of DPC and SNNDPC against the IK and Euclidean variants when the units/scales used to measure/represent the data features vary. As we have already shown that MP and Lin have similar performance, we do not include the results of Lin in this section. We use six datasets (Jain, WDBC, Parkinson, Thyroid, Libra and Gtzan) having different characteristics (size, dimensions, clusters and applications) for this experiment. We note that for the sake of brevity, we avoid presenting the results of the other datasets.

For the comparison, we transformed the data values of each feature x to: $\sqrt{x}$, $\log x$ and x^{-1}. Note that the transformations $\log x$ and $1/x$ are undefined for $x = 0$, Therefore, we apply the transformation to $c(x+\alpha)$, where $\alpha = 0.0001$ and $c = 100$ to obtain the results as done in [13]. The scaled data is again normalised in the range of [0,1].

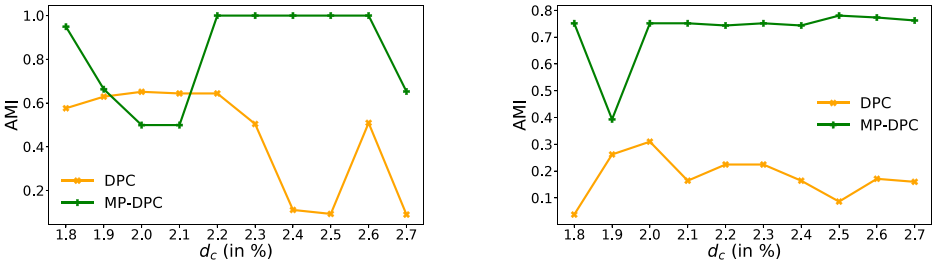
Figure 8 shows the bar plots of the best AMI score achieved by contending algorithms in six different datasets. As can be observed from the plots of different datasets, the AMI scores of DPC, SNNDPC, IK-DPC and IK-SNNDPC varies greatly when the scale is changed. On the other hand, the results of MP-DPC and MP-SNNDPC are not affected on the $\sqrt{x}$ and $\log x$ scale. A slight variation is noticed in the *inverse* transformation for the MP variants of DPC and SNNDPC because of the binning involved in the preprocessing step where bin cut points falls in bins at different sides as inverse reverses the order. These results show that MP-DPC and MP-SNNDPC are robust to changes in the scale of the data. The above results have been previously hypothesized and shown in Section 4.

To further explain these results, we visualise the effect of varying scale on the results of the DPC, SNNDPC and MP-DPC using the Jain and Thyroid datasets. We plot the clusters obtained for these datasets using the above algorithms on the x, $\sqrt{x}$ and $\log x$ scale and check if there is a variation. Tables 6 and 7 show the plots of true clusters based on ground truth (in the first row) and clusters identified by DPC, SNNDPC and MP-DPC in the Jain and Thyroid datasets. To plot the results of the Thyroid dataset, we reduced its dimensionality to 2 using PCA. Note that we have not used all the algorithms (such as IK-based DPC or SNNPDC and MP-SNNDPC) or scales (x^{-1}), as we have already covered them in Fig. 8.

As observed in both Tables 6 and 7, MP-DPC produces clusters very similar to the ground truths in all cases, whereas those of DPC and SNNDPC vary significantly. In particular, DPC and SNNDPC results are worst on the Thyroid dataset with scaled features. These results of DPC and SNNDPC confirm our previous observations. The near perfect clustering results of MP-DPC are due to its ability to adapt to the local density distribution and robustness to the scaling of data features.

5.2.3. Sensitivity to parameter d_c

In this section, we analyse the sensitivity of the parameter d_c in DPC and MP-DPC. We used the Jain (synthetic) and Thyroid (real-world) datasets for this purpose and compare the effect of varying



(a) Jain dataset. (b) Thyroid Dataset.

Fig. 9. Effect of varying d_c.

d_c on the clustering results of MP-DPC and DPC. Ten d_c values corresponding to different percentages in the range of 1% to 3% were chosen as shown in Fig. 9. For the Jain dataset, MP-DPC showed perfectly consistent results for values of d_c from 2.2 to 2.6. DPC, on the other hand, had highly varying results in the set of selected values. Similarly for the Thyroid dataset, while MP-DPC achieved similar AMI score for many d_c values, the results of DPC varied for the different values of d_c as shown in the figure.

Overall, the results show that MP-DPC is less sensitive to the change in the value of d_c compared to DPC with Euclidean distance.

6. Conclusion

In this paper, we focused on the popular Density Peak Clustering (DPC) algorithm, which is used in many applications. However, its inability to deal with varying density clusters and sensitivity to the representation of data limit its effectiveness for real-world problems where: (i) data have complex structure with varying density clusters; and (ii) how data features are expressed/represented may not be known. We overcome these two limitations of DPC by introducing a new data-dependent and scale-invariant similarity measure, which we call MP-Similarity. We show that MP-Similarity when incorporated into DPC (i) improves its performance on a range of datasets with varying characteristics, and (ii) provides consistent results even when the data is represented using different scales. Similar improvement in performance is also observed when MP-Similarity is used with SNNDPC algorithm, a variant of DPC algorithm, thus demonstrating the advantage of using MP-Similarity. Further, we show that another data-dependent and scale-invariant similarity measure (known as Lin's measure) also improves the performance of DPC and SNNDPC algorithm. This demonstrates the effectiveness of such measures in clustering complex real-world data. The experimental results using the popular clustering metrics validate our claim.

Although MP-Similarity produces better results, it has a few limitations. Because MP-Similarity is based on m_0-dissimilarity, which assumes that the attributes are independent and computes the dissimilarity in each dimension separately, it may perform poorly when there is a strong correlation between the attributes. In such cases, measures such as IK may perform better. However, noting from the experimental insights, MP-DPC provides better performance compared to other contenders across various datasets of different dimensions and sizes, which shows that such effect may not be significant in practice.

Future work includes investigating the performance of MP-Similarity on other popular clustering algorithms such as DBSCAN and hierarchical clustering. These algorithms also use (dis)similarity measures based on data-independent Euclidean distance. We also plan to extend this work on mixed data consisting of both numerical and categorical attributes as these are common for many real-world datasets. It would also be of interest to study the behaviour of the proposed similarity in the context of classification and anomaly detection.

Declaration of Competing Interest

The author declare that they have no known competing financial interests or personal relationship that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This research is funded by the US Air Force Office of Scientific Research (AFOSR) and Office of Naval Research (ONR) Global under grant number FA2386-20-1-4005.

References

[1] E. Diday, G. Govaert, Y. Lechevallier, J. Sidi, Clustering in pattern recognition, in: J.C. Simon, R.M. Haralick (Eds.), Digital Image Processing, Springer Netherlands, Dordrecht, 1981, pp. 19–58.
[2] Q. Zou, G. Lin, X. Jiang, X. Liu, X. Zeng, Sequence clustering in bioinformatics: an empirical study, Brief. Bioinformatics 21 (1) (2020) 1–10.
[3] J. Hou, W. Liu, E. Xu, H. Cui, Towards parameter-independent data clustering and image segmentation, Pattern Recognit. 60 (2016) 25–36.
[4] M.R. Bouadjenek, S. Sanner, Y. Du, Relevance-and interface-driven clustering for visual information retrieval, Inf. Syst. 94 (2020) 101592.
[5] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[6] D. Kobak, W. Brendel, C. Constantinidis, C.E. Feierstein, A. Kepecs, Z.F. Mainen, X.-L. Qi, R. Romo, N. Uchida, C.K. Machens, Demixed principal component analysis of neural population data, Elife 5 (2016) e10989.
[7] K. Sun, X. Geng, L. Ji, Exemplar component analysis: a fast band selection method for hyperspectral imagery, IEEE Geosci. Remote Sens. Lett. 12 (5) (2014) 998–1002.
[8] S. Zamuner, A. Rodriguez, F. Seno, A. Trovato, An efficient algorithm to perform local concerted movements of a chain molecule, PLoS ONE 10 (3) (2015) e0118342.
[9] W. Wang, J. Shen, F. Porikli, R. Yang, Semi-supervised video object segmentation with super-trajectories, IEEE Trans. Pattern Anal. Mach. Intell. 41 (4) (2018) 985–998.
[10] T. Anwar, C. Liu, H.L. Vu, C. Leckie, Partitioning road networks using density peak graphs: efficiency vs. accuracy, Inf. Syst. 64 (2017) 22–40.
[11] K.M. Dean, L.M. Davis, J.L. Lubbeck, P. Manna, P. Friis, A.E. Palmer, R. Jimenez, High-speed multiparameter photophysical analyses of fluorophore libraries, Anal. Chem. 87 (10) (2015) 5026–5030.
[12] Y. Zhang, Y. Xia, Y. Liu, W. Wang, Clustering sentences with density peaks for multi-document summarization, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015, pp. 1262–1267.
[13] S. Aryal, K.M. Ting, T. Washio, G. Haffari, A comparative study of data-dependent approaches without learning in measuring similarities of data objects, Data Min. Knowl. Discov. 34 (1) (2020) 124–162.
[14] R. Liu, H. Wang, X. Yu, Shared-nearest-neighbor-based clustering by fast search and find of density peaks, Inf. Sci. 450 (2018) 200–226.
[15] D. Xu, Y. Tian, A comprehensive survey of clustering algorithms, Ann. Data Sci. 2 (2) (2015) 165–193.

[16] S. Lloyd, Least squares quantization in PCM, IEEE Trans. Inf. Theory 28 (2) (1982) 129–137.
[17] P.H. Sneath, R.R. Sokal, et al., Numerical taxonomy, Nature 193 (4818) (1962) 855–860.
[18] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of KDD, 1996, pp. 226–231.
[19] G.J. McLachlan, T. Krishnan, The EM Algorithm and Extensions, Vol. 382, John Wiley & Sons, 2007.
[20] W. Wang, J. Yang, R. Muntz, et al., Sting: a statistical information grid approach to spatial data mining, in: VLDB, Vol. 97, Citeseer, 1997, pp. 186–195.
[21] J. Xie, H. Gao, W. Xie, X. Liu, P.W. Grant, Robust clustering by detecting density peaks and assigning points based on fuzzy weighted k-nearest neighbors, Inf. Sci. 354 (2016) 19–40.
[22] M. Du, S. Ding, H. Jia, Study on density peaks clustering based on k-nearest neighbors and principal component analysis, Knowl. Based Syst. 99 (2016) 135–145.
[23] J. Hou, A. Zhang, N. Qi, Density peak clustering based on relative density relationship, Pattern Recognit. 108 (2020) 107554.
[24] Y. Wang, D. Wang, X. Zhang, W. Pang, C. Miao, A.-H. Tan, Y. Zhou, McDPC: multi-center density peak clustering, Neural Comput. Appl. 32 (17) (2020) 13465–13478.
[25] A. Lotfi, P. Moradi, H. Beigy, Density peaks clustering based on density backbone and fuzzy neighborhood, Pattern Recognit. 107 (2020) 107449.
[26] M. Abbas, A. El-Zoghabi, A. Shoukry, DenMune: density peak based clustering using mutual nearest neighbors, Pattern Recognit. 109 (2021) 107589.
[27] L. Bai, X. Cheng, J. Liang, H. Shen, Y. Guo, Fast density clustering strategies based on the k-means algorithm, Pattern Recognit. 71 (2017) 375–386.
[28] Z. Rasool, R. Zhou, L. Chen, C. Liu, J. Xu, Index-based solutions for efficient density peak clustering, IEEE Trans. Knowl. Data Eng. (2020).
[29] A. Tversky, Features of similarity, Psychol. Rev. 84 (4) (1977) 327.
[30] C.L. Krumhansl, Concerning the applicability of geometric models to similarity data: the interrelationship between similarity and spatial density, Psychol. Rev. (1978).
[31] K.M. Ting, Y. Zhu, Z.-H. Zhou, Isolation kernel and its effect on SVM, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2329–2337.
[32] D. Lin, An information-theoretic definition of similarity, in: Proceedings of the Fifteenth International Conference on Machine Learning, in: ICML '98, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998, pp. 296–304.
[33] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 Eighth IEEE International Conference on Data Mining, IEEE, 2008, pp. 413–422.
[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[35] D. Arthur, S. Vassilvitskii, k-means++: The Advantages of Careful Seeding, Technical Report, Stanford, 2006.
[36] D. Dua, C. Graff, UCI machine learning repository, 2019, (http://archive.ics.uci.edu/ml).
[37] P. Fränti, M. Rezaei, Q. Zhao, Centroid index: cluster level similarity measure, Pattern Recognit. 47 (9) (2014) 3034–3045.
[38] E.B. Fowlkes, C.L. Mallows, A method for comparing two hierarchical clusterings, J. Am. Stat. Assoc. 78 (383) (1983) 553–569.
[39] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

Zafaryab Rasool is an Associate Research Fellow in the School of Information Technology, Deakin University, Australia. He completed his PhD in Computer Science from the Swinburne University of Technology, Australia in 2021. Previously, he earned his Master's degree from the Jamia Millia Islamia University, India in 2016 and Bachelor's degree from the Aligarh Muslim University, India in 2013. His research interest is in the areas of Data Mining, Machine Learning and Database, particularly focused on the clustering techniques.

Dr. Sunil Aryal is a Senior Lecturer in the School of Information Technology, Deakin University, Australia. Prior to joining Deakin in 2019, he worked as a lecturer at Federation University and worked in industry as a software developer and data analyst. He received his PhD from Monash University, Australia in 2017. His research interests are in the areas of Data Mining (DM), Machine Learning (ML), and Artificial Intelligence (AI), particularly in their applications to solve real-world problems. He has published more than 40 scientific papers in top tier DM/ML conferences/journals. His research is supported by US and Australian Defence and Intelligence agencies. He has been an investigator on research grants/contracts with funding over $2.3 millions.

Mohamed Reda Bouadjenek, PhD: Reda is a Lecturer in the School of Information Technology at Deakin University, Geelong, Australia. Previously, he was a Research Fellow at The University of Toronto (2017–2019) and at The University of Melbourne (2015–2017) and before that, he was a postdoc researcher at INRIA France (2014–2015). Reda earned a PhD and an MSc in Computer Science from the University of Paris-Saclay France respectively in 2013 and 2009, and a BSc in Computer Science from USTHB Algeria in 2008. His research spans a broad range of topics related to the data-driven fields of Machine Learning, Deep Learning, and Information Retrieval. He has applied analytic and algorithmic tools from these fields to solve real-world problems related to diverse applications such as recommender systems, interactive visual search interfaces, social network analysis, and data quality.

Richard Dazeley is an Associate Professor of Computer Science at Deakin University (Geelong). He is a highly experienced researcher in multiobjective, interactive, deep, safe and explainable Reinforcement Learning (RL). He has published several of the most widely cited papers in multiobjective RL where his work significantly contributed to the foundation of this field of research. He been recognized as a nationally leading ICT Curriculum Designer - receiving the Australian National Award as ICT Educator of the Year (2016) from the ACS and was nominated for the International ICT Educator of the Year award from the South-East Asia Regional Computer Confederation (SEARCC) (2017). Prior to working at Deakin, he was employed at Federation University Australia where he was the Head of the IT Discipline and the co-Founder and Leader of the Federation Learning Agents Research Group (FLAG).