A New Type of Distance Metric and Its Use for Clustering

Xiaowei Gu[1], Plamen Angelov[1], Dmitry Kangin[1] and Jose Principe[2]

[1] School of Computing and Communications, Lancaster University Lancaster, LA1 4WA, UK. e-mail: {x.gu3, p.angelov, d.kangin}@lancaster.ac.uk

[2] Computational NeuroEngineering Laboratory, Department of Electrical and Computer Engineering, University of Florida, USA. e-mail: principe@cnel.ufl.edu

Abstract- In order to address high dimensional problems, a new 'direction-aware' metric is introduced in this paper. This new distance is a combination of two components: i*)* the traditional Euclidean distance and *ii)* an angular/directional divergence, derived from the cosine similarity. The newly introduced metric combines the advantages of the Euclidean metric and cosine similarity, and is defined over the Euclidean space domain. Thus, it is able to take the advantage from both spaces, while preserving the Euclidean space domain. The direction-aware distance has wide range of applicability and can be used as an alternative distance measure for various traditional clustering approaches to enhance their ability of handling high dimensional problems. A new evolving clustering algorithm using the proposed distance is also proposed in this paper. Numerical examples with benchmark datasets reveal that the direction-aware distance can effectively improve the clustering quality of the k-means algorithm for high dimensional problems and demonstrate the proposed evolving clustering algorithm to be an effective tool for high dimensional data streams processing.

Index Terms- cosine similarity; distance metric; metric space; clustering; high dimensional data streams processing.

1. Introduction

The widely used clustering techniques may use different kind of distances to measure the separation between data samples. The well-known Euclidean distance is currently the most frequently used metric space for the established clustering algorithms [1], [2]. Other metric spaces, using the Mahalanobis [3], city block Hamming, Minkowski types of distances, etc., are also widely used in different clustering algorithms for different purposes. It is often the case that clustering algorithms employing divergences, i.e. pairwise dissimilarity, which does not obey all the properties of distances (e.g. cosine similarity), could generate meaningless conclusions.

One problem the traditional distance metrics are facing is the so-called "curse of dimensionality" [4], [5]. Many clustering techniques, which use the traditional distance metrics work well in low dimensional space, however, become intractable for high dimensional problems. Research results have shown that in high dimensional space, the concept of distance may not even be qualitatively meaningful [5], [6]. This phenomenon is frequently seen in the cases that some dimensions of the data are highly irrelevant. This is not hard to understand because our intuitions come from a three-dimensional world only, which may not be applicable to high dimensional ones.

Compared with the commonly used distance metrics including the Euclidean, Mahalanobis, Minkowski distances, etc., which measure the magnitude of vector difference, cosine similarity focuses much more on the directional similarity. Therefore, it is more often used in the natural language processing (NLP) problems [7]–[11]. In NLP problems, machine learning algorithms, for example, *k-means* [7], [10], *mean shift* [11], etc., are used to cluster very high dimensional vectors representing the documents together based on the cosine similarity. Nonetheless, the cosine similarity is a pseudo metric because it does not obey the triangle inequality (it obeys the Cauchy-Schwarz inequality [12]). Consequently, the cosine similarity between two vectors can be

In this paper, a new "direction-aware" distance is introduced. This new metric space is a combination of a distance (in this paper, we consider Euclidean), and an angular/directional component, which is based on the cosine similarity, where the weights of the Euclidean and angular components are under the user control. Therefore, it takes the advantages of the both components while still obeys all the properties of a distance metric [13] as we will demonstrate.

The proposed distance in this paper is applicable to various traditional clustering algorithms as an alternative distance measure and can enhance the ability of the algorithms to handle high dimensional problems. A new evolving clustering algorithm is also proposed for streaming data processing. This algorithm employs the new direction-aware distance only and is able to start from scratch. Therefore, it is very suitable for handling the high dimensional data streams.

Numerical examples using benchmark datasets demonstrate the potential of the direction-aware distance against many traditional metrics in high dimensional problems. It is also shown that the proposed clustering algorithm is able to produce top quality clustering results on various problems with high computational efficiency.

The remainder of this paper is organised as follows. Section 2 describes the newly proposed direction-aware distance and provides the proof for the proposed distance to be a full metric. Section 3 introduces the application of the newly proposed direction-aware distance to traditional clustering algorithms. The new evolving clustering algorithm based on the proposed distance is presented in section 4. Section 5 presents numerical examples. The paper is concluded by section 6.

2. Direction-Aware Distance and Proof of Metric Axioms

A. The New Direction-Aware Distance

In this section, we introduce the direction-aware distance, and prove that it is a distance over the space of real numbers. If no specific declaration is provided, all the derivations in this paper are conducted over the real numbers.

First of all, let us define a metric space, $\mathbf{R}^m$, $\boldsymbol{x}$ and $\boldsymbol{y}$ are two data points within the space, m is the dimensionality of the metric space $\mathbf{R}^m$. The newly introduced direction-aware distance, $d_{DA}(\boldsymbol{x},\boldsymbol{y})$ consists of two terms:

i) a Euclidean component, $d_M(\boldsymbol{x},\boldsymbol{y})$, and

ii) a direction-aware component, $d_A(\boldsymbol{x},\boldsymbol{y})$,

and is expressed as:

$$d_{DA}(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\lambda_M^2\left(d_M(\boldsymbol{x},\boldsymbol{y})\right)^2 + \lambda_A^2\left(d_A(\boldsymbol{x},\boldsymbol{y})\right)^2} \tag{1}$$

where $\boldsymbol{x} = [x_1, x_2, ..., x_m]^{\mathrm{T}}$ and $\boldsymbol{y} = [y_1, y_2, ..., y_m]^{\mathrm{T}}$, $\boldsymbol{x}, \boldsymbol{y} \in \mathbf{R}^m$; λ_M, λ_A are a pair of scaling coefficients and $\lambda_M > 0$, $\lambda_A > 0$; $d_M(\boldsymbol{x},\boldsymbol{y})$ denotes the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{y}$, $d_M(\boldsymbol{x},\boldsymbol{y}) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{y})}$ $= \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}$.

The direction-aware component $d_A(\boldsymbol{x},\boldsymbol{y})$ is derived based on the cosine similarity expressed by:

$$d_A(\boldsymbol{x},\boldsymbol{y}) = \sqrt{1 - \cos(\Theta_{xy})} \tag{2}$$

where Θ_{xy} is the angle between $\boldsymbol{x}$ and $\boldsymbol{y}$. In the Euclidean space, since $\langle \boldsymbol{x},\boldsymbol{y} \rangle = \sum_{i=1}^{m} x_i y_i$ and $\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x},\boldsymbol{x} \rangle}$, thus $\cos(\Theta_{xy}) = \frac{\langle \boldsymbol{x},\boldsymbol{y} \rangle}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}$. Therefore, the directional component $d_A(\boldsymbol{x},\boldsymbol{y})$ can be rewritten as:

$$d_A(\boldsymbol{x},\boldsymbol{y}) = \sqrt{1 - \frac{\langle \boldsymbol{x},\boldsymbol{y} \rangle}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}} = \sqrt{1 - \frac{\sum_{i=1}^{m} x_i y_i}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}} = \sqrt{\frac{\sum_{i=1}^{m} x_i^2}{2\|\boldsymbol{x}\|^2} + \frac{\sum_{i=1}^{m} y_i^2}{2\|\boldsymbol{y}\|^2} - \frac{\sum_{i=1}^{m} x_i y_i}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}} = \sqrt{\frac{1}{2}\sum_{i=1}^{m}\left(\frac{x_i}{\|\boldsymbol{x}\|} - \frac{y_i}{\|\boldsymbol{y}\|}\right)^2} = \frac{1}{\sqrt{2}}\left\|\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|} - \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|}\right\| \tag{3}$$

One can notice that, if $\boldsymbol{x}$ or $\boldsymbol{y}$ is equal to $\mathbf{0}$, $d_A(\boldsymbol{x},\boldsymbol{y}) = 0$.

B. Proof of Metric Axioms

In this subsection, we will prove that the proposed distance is a full metric. For a distance $d(\boldsymbol{x},\boldsymbol{y})$ in the space to be a full metric, $\mathbf{R}^m$, it is required to satisfy the following properties for $\forall \boldsymbol{x},\boldsymbol{y}$ [13]:

i) non-negativity: $d(\boldsymbol{x},\boldsymbol{y}) \geq 0$; (4)

ii) identity of indiscernibles: $d(\boldsymbol{x},\boldsymbol{y})=0$ *iff* $\boldsymbol{x}=\boldsymbol{y}$; (5)

iii) symmetry: $d(\boldsymbol{x},\boldsymbol{y})=d(\boldsymbol{y},\boldsymbol{x})$; (6)

iv) triangle inequality: $d(\boldsymbol{x},\boldsymbol{z})+d(\boldsymbol{z},\boldsymbol{y})\geq d(\boldsymbol{x},\boldsymbol{y})$. (7)

In this paper, we propose a new theorem as follows:

Theorem: $d_{DA}(\boldsymbol{x},\boldsymbol{y})$ is a distance within the metric space over the domain $\mathbf{R}^m$.

In the rest of this subsection, we will prove this theorem by proving that $d_{DA}(\boldsymbol{x},\boldsymbol{y})$ obeys the four distance axioms stated in equations (5)-(6) and inequalities (4) and (7) one by one.

Lemma 1: $\forall \boldsymbol{x},\boldsymbol{y}\in\mathbf{R}^m$, $d_{DA}(\boldsymbol{x},\boldsymbol{y})\geq 0$.

Proof: It can be seen directly from the equation (5) that $d_{DA}(\boldsymbol{x},\boldsymbol{y})$ is always non-negative.

Lemma 2: $\forall \boldsymbol{x},\boldsymbol{y}\in\mathbf{R}^m$, $d_{DA}(\boldsymbol{x},\boldsymbol{y})=0$ *iff* $\boldsymbol{x}=\boldsymbol{y}$.

Proof: It is clear that if $\boldsymbol{x}=\boldsymbol{y}$, then $d_A(\boldsymbol{x},\boldsymbol{y})=\sqrt{1-1}=0$, $d_M(\boldsymbol{x},\boldsymbol{y})=0$ and $d_{DA}(\boldsymbol{x},\boldsymbol{y})=0$.

The directional component $d_A(\boldsymbol{x},\boldsymbol{y})$ alone does not obey this property because as we can see from equations (2) and (3), if $\boldsymbol{x}$ and $\boldsymbol{y}$ are nonzero and orthogonal, $d_A(\boldsymbol{x},\boldsymbol{y})=0$, so it is not true. However, in this case due to the fact that if $\boldsymbol{x}\neq\boldsymbol{y}$, $d_M(\boldsymbol{x},\boldsymbol{y})\neq 0$, $d_{DA}(\boldsymbol{x},\boldsymbol{y})$ will still be non-zero as $\lambda_M,\lambda_A>0$. Therefore, one can still conclude that $d_{DA}(\boldsymbol{x},\boldsymbol{y})=0$ if and only if $\boldsymbol{x}=\boldsymbol{y}$.

Lemma 3: $\forall \boldsymbol{x},\boldsymbol{y}\in\mathbf{R}^m$, $d_{DA}(\boldsymbol{x},\boldsymbol{y})=d_{DA}(\boldsymbol{y},\boldsymbol{x})$

Proof: For the Euclidean metric, it is true that:

$$d_{DA}(\boldsymbol{x},\boldsymbol{y})=\sqrt{\lambda_M^2\left\|\boldsymbol{x}-\boldsymbol{y}\right\|^2+\frac{\lambda_A^2}{2}\left\|\frac{\boldsymbol{x}}{\left\|\boldsymbol{x}\right\|}-\frac{\boldsymbol{y}}{\left\|\boldsymbol{y}\right\|}\right\|^2}=\sqrt{\lambda_M^2\left\|\boldsymbol{y}-\boldsymbol{x}\right\|^2+\frac{\lambda_A^2}{2}\left\|\frac{\boldsymbol{y}}{\left\|\boldsymbol{y}\right\|}-\frac{\boldsymbol{x}}{\left\|\boldsymbol{x}\right\|}\right\|^2}=d_{DA}(\boldsymbol{y},\boldsymbol{x}) \tag{8}$$

Therefore, $d_{DA}(\boldsymbol{x},\boldsymbol{y})=d_{DA}(\boldsymbol{y},\boldsymbol{x})$.

Lemma 4: $\forall \boldsymbol{x},\boldsymbol{y},\boldsymbol{z}\in\mathbf{R}^m$, $d_{DA}(\boldsymbol{x},\boldsymbol{z})\leq d_{DA}(\boldsymbol{x},\boldsymbol{y})+d_{DA}(\boldsymbol{y},\boldsymbol{z})$

Proof: Firstly, let us assume that there is a triplet data samples $\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}$, which make d_{DA} break the triangle rule, namely:

$$d_{DA}(\boldsymbol{x},\boldsymbol{z})>d_{DA}(\boldsymbol{x},\boldsymbol{y})+d_{DA}(\boldsymbol{y},\boldsymbol{z}) \tag{9}$$

By including equation (3) in equation (2), the direction-aware distance $d_{DA}(\boldsymbol{x},\boldsymbol{y})$ can be rewritten as:

$$\begin{aligned} d_{DA}(\boldsymbol{x},\boldsymbol{y})&=\sqrt{\lambda_M^2\left(d_M(\boldsymbol{x},\boldsymbol{y})\right)^2+\lambda_A^2\left(d_A(\boldsymbol{x},\boldsymbol{y})\right)^2}=\sqrt{\lambda_M^2\left\|\boldsymbol{x}-\boldsymbol{y}\right\|^2+\frac{\lambda_A^2}{2}\left\|\frac{\boldsymbol{x}}{\left\|\boldsymbol{x}\right\|}-\frac{\boldsymbol{y}}{\left\|\boldsymbol{y}\right\|}\right\|^2} \\ &=\sqrt{\lambda_M^2\sum_{i=1}^{m}\left(x_i-y_i\right)^2+\frac{\lambda_A^2}{2}\sum_{i=1}^{m}\left(\frac{x_i}{\left\|\boldsymbol{x}\right\|}-\frac{y_i}{\left\|\boldsymbol{y}\right\|}\right)^2}=\left\|\boldsymbol{\chi}-\boldsymbol{\psi}\right\|=d_M(\boldsymbol{\chi},\boldsymbol{\psi}) \end{aligned} \tag{10}$$

where, $\boldsymbol{\chi}=\left[\lambda_M\boldsymbol{x}^{\mathrm{T}},\frac{\lambda_A\boldsymbol{x}^{\mathrm{T}}}{\sqrt{2}\left\|\boldsymbol{x}\right\|}\right]^{\mathrm{T}}=\left[\lambda_M x_1,\lambda_M x_2,...,\lambda_M x_m,\frac{\lambda_A x_1}{\sqrt{2}\left\|\boldsymbol{x}\right\|},\frac{\lambda_A x_2}{\sqrt{2}\left\|\boldsymbol{x}\right\|},...,\frac{\lambda_A x_m}{\sqrt{2}\left\|\boldsymbol{x}\right\|}\right]^{\mathrm{T}}$ and

$$\boldsymbol{\psi} = \left[\lambda_M \boldsymbol{y}^{\mathrm{T}}, \frac{\lambda_A \boldsymbol{y}^{\mathrm{T}}}{\sqrt{2}\|\boldsymbol{y}\|} \right]^{\mathrm{T}} = \left[\lambda_M y_1, \lambda_M y_2, ..., \lambda_M y_m, \frac{\lambda_A y_1}{\sqrt{2}\|\boldsymbol{y}\|}, \frac{\lambda_A y_2}{\sqrt{2}\|\boldsymbol{y}\|}, ..., \frac{\lambda_A y_m}{\sqrt{2}\|\boldsymbol{y}\|} \right]^{\mathrm{T}} .$$

Similarly, for $\boldsymbol{\varsigma} = \left[\lambda_M \boldsymbol{z}^{\mathrm{T}}, \frac{\lambda_A \boldsymbol{z}^{\mathrm{T}}}{\sqrt{2}\|\boldsymbol{z}\|} \right]^{\mathrm{T}}$, we can see that $d_{DA}(\boldsymbol{x},\boldsymbol{z}) = d_M(\boldsymbol{\chi},\boldsymbol{\varsigma})$, $d_{DA}(\boldsymbol{y},\boldsymbol{z}) = d_M(\boldsymbol{\psi},\boldsymbol{\varsigma})$.

Considering an auxiliary algebraic data space $\mathbf{R}^{2m}$, for $\boldsymbol{\chi},\boldsymbol{\psi},\boldsymbol{\varsigma}$, it follows that:

$$d_M(\boldsymbol{\chi},\boldsymbol{\varsigma}) \le d_M(\boldsymbol{\chi},\boldsymbol{\psi}) + d_M(\boldsymbol{\psi},\boldsymbol{\varsigma}) \tag{11}$$

As we can see from inequalities (9) and (11), the two equations have the same algebraic form, but there are different signs ($>$ and $\le$). For Euclidean distance in $\mathbf{R}^{2m}$, the triangle rule is always conformed, therefore, we can conclude that $d_{DA}(\boldsymbol{x},\boldsymbol{y})$ always satisfies the triangle inequality: $d_{DA}(\boldsymbol{x},\boldsymbol{z}) \le d_{DA}(\boldsymbol{x},\boldsymbol{y}) + d_{DA}(\boldsymbol{y},\boldsymbol{z})$.

Based on the proofs of the four **lemmas**, the proposed **Theorem** is proven. Therefore, we can conclude that the proposed *direction-aware* distance, d_{DA} is a full distance in the Euclidean space.

C. The Property of the Proposed Distance

The proposed direction-aware distance metric is a combination of two components: *i)* the traditional Euclidean distance and *ii)* an angular/directional divergence, derived from the cosine similarity. It defines a metric space as a combination of Euclidean metric space and cosine similarity pseudo-metric space, and consequently, can effectively combine information extracted from both spaces and takes into account both spatial and angular divergences. Therefore, the direction aware distance can serve as a more representative distance metric than the traditional distance metric.

3. The Application of the Proposed Distance to Traditional Clustering Approaches

In this section, we will describe the applications of the proposed distance to the traditional offline clustering approaches. First of all, let us define the dataset in the metric space as $\{\boldsymbol{x}\}_N = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\} \in \mathbf{R}^m$, $\boldsymbol{x}_i = [x_{i,1}, x_{i,2}, ..., x_{i,d}]^{\mathrm{T}} \in \mathbf{R}^d$, $i = 1, 2, ..., N$, where N is the number of data samples in the dataset.

The newly proposed direction-aware distance can be used in various clustering, classification as well as regression approaches. For example, the k-means [7], [10], mean-shift clustering [14], k nearest neighbour classification [15] algorithms may use the newly introduced direction-aware distance to enhance the ability in dealing with high dimensional data.

Since the traditional offline algorithms have been studied well for many years, in this paper, we will not focus on the algorithm themselves. Instead, we will look at the direction-aware distance and introduce the strategy of using the proposed distance in the algorithms for different purposes.

The direction-aware distance has a pair of scaling factors, the values of which can be adjusted for various problems. For example, if without losing generality, we want to allocate the same importance to the Euclidean and directional components, λ_M and λ_A can be set as the inverse of average d_M and d_A, respectively (the data is taken without pre-processing):

$$\lambda_M = \frac{1}{\sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N} d_M^2(\boldsymbol{x}_i,\boldsymbol{x}_j)}{N^2}}} \tag{12a}$$

$$\lambda_A = \frac{1}{\sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N} d_A^2(\boldsymbol{x}_i,\boldsymbol{x}_j)}{N^2}}} \tag{12b}$$

Alternatively, if the data has been re-scaled to the range $[0,1]$ in advance, the values of d_M and d_A are within the ranges $\left[0,\sqrt{m}\right]$ and $[0,1]$, respectively, thus, the pair of the scaling coefficients within the proposed distance can be set to $\lambda_M = \frac{1}{\sqrt{m}}$ and $\lambda_A = 1$ if we aim to allocate the same importance to each component.

While for some problems like NLP, where the directional similarity plays a more important role compared with magnitude differences, we can enhance the importance of the directional component in the distance measures by increasing the value of λ_A , and vice versa.

4. The Applications of the Proposed Distance to Evolving Clustering

Similarly, the direction-aware distance can also be employed in the evolving clustering approaches. In this section, we propose a new evolving clustering approach with the direction-aware distance. This algorithm is able to "start from scratch" and consistently evolve its system structure and update the meta-parameters based on the newly arrived data samples.

The main procedure of the proposed algorithm is described as follows. In this section, we consider $\{\boldsymbol{x}\}_k = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k\} \in \mathbf{R}^m$ as a data stream and the subscript indicates the time instance that the data sample arrives.

Stage 1. Initialization

The first data sample $\boldsymbol{x}_1$ in the data stream is used for initializing the system and its meta-parameters. In the proposed algorithm, the system has the following initialized global meta-parameters:

i. $k \leftarrow 1$, the current time instance;

ii. $C \leftarrow 1$, the number of exiting clusters;

iii. $\boldsymbol{\mu}_M \leftarrow \boldsymbol{x}_1$, the global mean of $\{\boldsymbol{x}\}_k$;

iv. $X_M \leftarrow \|\boldsymbol{x}_1\|^2$, the global average scalar product of $\{\boldsymbol{x}\}_k$;

v. $\boldsymbol{\mu}_A \leftarrow \frac{\boldsymbol{x}_1}{\|\boldsymbol{x}_1\|}$, the global mean of $\left\{\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}\right\}_k$, which is also the normalized global mean of $\{\boldsymbol{x}\}_k$.

vi. $X_A \leftarrow \left\|\frac{\boldsymbol{x}_1}{\|\boldsymbol{x}_1\|}\right\|^2 = 1$, the global average scalar product of $\left\{\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}\right\}_k$, which is always equal to 1.

The local meta-parameters of the first cluster are initialized as follows:

i. $\Xi^1 \leftarrow \{\boldsymbol{x}_1\}$, the first cluster;

ii. $\boldsymbol{f}_M^1 \leftarrow \boldsymbol{x}_1$, the centre of the first cluster, which is also the mean of Ξ^1 ;

iii. $X_M^1 \leftarrow \|\boldsymbol{x}_1\|^2$, the average scalar product of Ξ^1 ;

iv. $\boldsymbol{f}_A^1 \leftarrow \frac{\boldsymbol{x}_1}{\|\boldsymbol{x}_1\|}$, the normalized mean of Ξ^1 ;

v . $X_A^1 \leftarrow 1$, the normalized average scalar product of Ξ^1 , which is always equal to 1 as well;

vi. $S^1 \leftarrow 1$, the support (population) of the first cluster.

After the initialization of the system, the proposed algorithm updates the system structure and meta-parameters with the arrival of each new data samples.

Stage 2. System Structure and Meta-Parameters Update

With each newly arrived data sample, the system's global meta-parameters, $\boldsymbol{\mu}_M$, X_M and $\boldsymbol{\mu}_A$ are updated using the following equations [16]:

i. $\boldsymbol{\mu}_M \leftarrow \frac{k}{k+1}\boldsymbol{\mu}_M + \frac{1}{k+1}\boldsymbol{x}_{k+1}$ (13a)

ii. $X_M \leftarrow \frac{k}{k+1}X_M + \frac{1}{k+1}\left\|\boldsymbol{x}_{k+1}\right\|^2$ (13b)

iii. $\boldsymbol{\mu}_A \leftarrow \frac{k}{k+1}\boldsymbol{\mu}_A + \frac{1}{k+1}\frac{\boldsymbol{x}_{k+1}}{\left\|\boldsymbol{x}_{k+1}\right\|}$ (13c)

iv. $k \leftarrow k+1$ (13d)

Then, the condition A is checked to see whether the new data sample denoted by $\boldsymbol{x}_k$ is associated with a new cluster:

Condition A: $IF\left(d_{DA}\left(\boldsymbol{x}_k,\boldsymbol{\mu}_M\right) > \max_{j=1,2,\ldots,C}\left(d_{DA}\left(\boldsymbol{f}_M^j,\boldsymbol{\mu}_M\right)\right)\right) OR \left(d_{DA}\left(\boldsymbol{x}_k,\boldsymbol{\mu}_M\right) < \min_{j=1,2,\ldots,C}\left(d_{DA}\left(\boldsymbol{f}_M^j,\boldsymbol{\mu}_M\right)\right)\right)$
$THEN\left(\boldsymbol{x}_k \text{ creates a new cluster}\right)$ (14)

Based on the previous subsection, without a loss of generality, we use the inverse of the average d_M between the existing data samples as λ_M and the inverse of the average d_A as λ_A, correspondingly. However, for streaming data processing, it is less efficient to keep all the observed data samples in the memory and recalculate λ_M and λ_A every time when a new data sample is observed. Therefore we introduce the recursive forms for calculating the pair of scaling coefficients as follows [16]:

$$\lambda_M = \frac{1}{\sqrt{\frac{\sum_{i=1}^{k}\sum_{j=1}^{k} d_M^2\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)}{k^2}}} = \frac{1}{\sqrt{2\left(X_M - \left\|\boldsymbol{\mu}_M\right\|^2\right)}} \tag{15a}$$

$$\lambda_A = \frac{1}{\sqrt{\frac{\sum_{i=1}^{k}\sum_{j=1}^{k} d_A^2\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)}{k^2}}} = \frac{1}{\sqrt{1-\left\|\boldsymbol{\mu}_A\right\|^2}} \tag{15b}$$

If condition A is satisfied, a new cluster is added with $\boldsymbol{x}_k$ as its centre:

i. $C \leftarrow C+1$, the number of existing clusters;

ii. $\Xi^C \leftarrow \{\boldsymbol{x}_k\}$, the new cluster;

iii. $\boldsymbol{f}_M^C \leftarrow \boldsymbol{x}_k$, the centre of the new cluster/ mean of Ξ^C;

iv. $X_M^C \leftarrow \left\|\boldsymbol{x}_k\right\|^2$, the average scalar product of Ξ^C;

v. $\boldsymbol{f}_A^C \leftarrow \frac{\boldsymbol{x}_k}{\left\|\boldsymbol{x}_k\right\|}$, the normalized centre of the new cluster/ normalized mean of Ξ^C;

vi. $X_A^C \leftarrow 1$, the normalized average scalar product of Ξ^C,

vii. $S^C \leftarrow 1$, the support of the new cluster.

In contrast, if condition A is not met, $\boldsymbol{x}_k$ is assigned to the cluster with the nearest centre, denoted by $\boldsymbol{f}_M^n$ as:

$$\boldsymbol{f}_M^n = \arg\min_{i=1,2,\ldots,C}\left(d_{DA}\left(\boldsymbol{x}_k,\boldsymbol{f}_M^i\right)\right) \tag{16}$$

The meta-parameters of the cluster with the nearest centre are updated as follows [16]:

i. $\Xi^n \leftarrow \Xi^n \cup \{\boldsymbol{x}_k\}$ (17a)

ii. $\boldsymbol{f}_M^n \leftarrow \frac{S^n}{S^n+1}\boldsymbol{f}_M^n + \frac{1}{S^n+1}\boldsymbol{x}_k$ (17b)

iii. $X_M^n \leftarrow \frac{S^n}{S^n+1}X_M^n + \frac{1}{S^n+1}\|\boldsymbol{x}_k\|^2$ (17c)

iv. $\boldsymbol{f}_A^n \leftarrow \frac{S^n}{S^n+1}\boldsymbol{f}_A^n + \frac{1}{S^n+1}\frac{\boldsymbol{x}_k}{\|\boldsymbol{x}_k\|}$ (17d)

v. $S^n \leftarrow S^n + 1$ (17e)

After the update of the global and local meta-parameters, the system is ready for the arrival of the next data sample and begins a new processing cycle.

Stage 3. Clusters Adjusting

In this stage, all the existing clusters will be examined and adjusted to avoid the possible overlap. For each existing cluster Ξ^i ($i = 1, 2, ..., C$), firstly, we find its neighbouring clusters, denoted by $\{\Xi\}_{neighbour}^i$ based on the following condition:

Condition B: $IF\left(d_{DA}\left(\boldsymbol{f}_M^i, \boldsymbol{f}_M^j\right) > \frac{\sum_{p=1}^{C}\sigma_{DA}^p}{C}\right)\ THEN\left(\{\Xi\}_{neighbour}^i \leftarrow \Xi^j \cup \{\Xi\}_{neighbour}^i\right)$ (18)

where $\left(\sigma_{DA}^p\right)^2 = \frac{\sum_{\boldsymbol{x}\in\Xi^p}\sum_{\boldsymbol{y}\in\Xi^p} d_{DA}^2(\boldsymbol{x},\boldsymbol{y})}{S_p^2} = 2\left(X_M^p - \|\boldsymbol{f}_M^p\|^2\right) + \left(1 - \|\boldsymbol{f}_A^p\|^2\right)$ is the average square direction-aware distance between all the members within the p^{th} cluster.

For each cluster centre, $\boldsymbol{f}_M^i$ ($i = 1, 2, ..., C$), we calculate its weighted unimodal density as [16]:

$$D^W\left(\boldsymbol{f}_M^i\right) = S^i \frac{\sum_{l=1}^{C}\sum_{j=1}^{C} d_{DA}^2\left(\boldsymbol{f}_M^l, \boldsymbol{f}_M^j\right)}{2C\sum_{j=1}^{C} d_{DA}^2\left(\boldsymbol{f}_M^i, \boldsymbol{f}_M^j\right)} \tag{19}$$

and we also compare $D^W\left(\boldsymbol{f}_M^i\right)$ with the D^W of its neighbouring clusters denoted by $\left\{D^W\left(\boldsymbol{f}_M\right)\right\}_{neighbour}^i$, to identify the local maxima of the weighted unimodal density, D^W:

Condition C: $IF\left(D^W\left(\boldsymbol{f}_M^i\right) > \max\left(\left\{D^W\left(\boldsymbol{f}_M\right)\right\}_{neighbour}^i\right)\right)$
$THEN\left(\boldsymbol{f}_M^i\ is\ one\ of\ the\ local\ maxiam\ of\ D^W\right)$ (20)

By identifying all the local maxima, denoted by $\{\boldsymbol{f}_M\}_o$ and assigning each data sample to the cluster with the nearest centre using equation (16), the whole clustering processing is finished. The parameters of the clusters can be extracted *post factum*.

The main procedure of the algorithm is summarised in the form of pseudo code as follows.

i. **While** a new data sample $\boldsymbol{x}_k$ of the data stream is available (or until interrupted)

*** If** (it is the first data sample) **Then**

- Initialise global meta-parameters: k, C, $\boldsymbol{\mu}_M$, X_M, $\boldsymbol{\mu}_A$, X_A;

- Initialise local meta-parameters of the first cluster: Ξ^1, $\boldsymbol{f}_M^1$, X_M^1, $\boldsymbol{f}_A^1$, X_A^1, S^1;

*** Else**

- Update $\boldsymbol{\mu}_M$, X_M, $\boldsymbol{\mu}_A$ and k using equation (13);

- **If** (Condition A is met) **Then**

1. $C \leftarrow C+1$;

2. Initialise local meta-parameters of the new cluster: Ξ^C, $\boldsymbol{f}_M^C$, X_M^C, $\boldsymbol{f}_A^C$, X_A^C, S^C;

- **Else**

1. Find the nearest cluster Ξ^n using equation (16);

2. Update the meta-parameter of this cluster using equation (17): Ξ^n, $\boldsymbol{f}_M^n$, X_M^n, $\boldsymbol{f}_A^n$, S^n.

- **End If**

* **End If**

ii. **End While**

iii. Find the neighbouring clusters $\{\Xi\}_n^i$ for each existing cluster Ξ^i using equation (18) ($i = 1, 2, ..., C$).

iv. Calculate the weighted unimodal densities at the centres of the clusters using equation (19);

v. Identify the local maxima of the weighted unimodal density using equation (20);

vi. Assign each data sample to the cluster with the nearest centre using equation (16).

5. Numerical Examples and Analysis

In this section, a number of numerical experiments are conducted to demonstrate the performance of the newly proposed direction-aware distance for high dimensional problems. Analysis based on the numerical examples will be provided.

Firstly, we use the standard k-means algorithm as a benchmark. We consider the following problems to test the performance of the k-means algorithm with different type of distance/similarity including Euclidean distance, cosine similarity, cityblock distance and the proposed direction-aware distance:

i) Dim256 dataset [17];

ii) Dim512 dataset [17];

iii) Dim1024 dataset [17];

iv) Dim15 dataset [17];

v) Steel plate faults dataset [18];

vi) Pen-based recognition of handwritten digits dataset [19];

vii) Optical recognition of handwritten digits dataset [20];

viii) Cardiotocography dataset [21];

The dim256, dim512, dim1024 and dim15 datasets are sampled from Gaussian distributions, and, thus, the four datasets are ideal for testing the ability of the algorithms in separating high dimensional data samples from different classes. The other 5 datasets are real benchmark problems and we use them to evaluate the performance of the algorithms on real, non-Gaussian problems. The details of the benchmark datasets are given in Table I.

TABLE I. Details of the Datasets

Abbreviation	Dataset	Samples	Classes	Attributes
D256	dim256	1024	16	256+1 label
D512	dim512	1024	16	512+1 label
D1024	dim1024	1024	16	1024+1 label
D15	dim15	10125	9	15+1 label
ST	Steel plates faults	1941	5	27+1 label
PE	Pen-based recognition	10992	10	16+1 label

OP	Optical recognition	5620	64	64+1 label
CA	Cardiotocography	2126	3	22+1 label

Because of the complexity of the high-dimensional problems, the clustering results of the *k-means* algorithm may exhibit some degree of randomness, for each dataset and each type of distance/similarity, we did 100 Monte Carlo experiments and tabulated the average values of the five different measures in Table II. The algorithms used in this paper were implemented within MATLAB 2015b; the performance was evaluated on a PC with dual core Intel i7 processor with clock frequency 3.4GHz each and 16 GB RAM. In the experiment, without loss of generality, the pair of the scaling parameters of the direction-aware distance is set by equation (12) and we consider the Calinski-Harabasz (CH) index [22] to evaluate the quality of the clustering results. Higher Calinski-Harabasz (CH) index indicates a better clustering quality.

TABLE II. Experimental Results

Dataset	Distance/Dissimilarity	*CH*	Dataset	Distance/Dissimilarity	*CH*
D256	Euclidean	405.2386	ST	Euclidean	20.2314
	Cosine	448.0036		Cosine	21.769
	Cityblock	424.2804		Cityblock	17.4560
	Direction-aware	**509.2634**		**Direction-aware**	**25.8675**
D512	Euclidean	373.8111	PE	Euclidean	575.0739
	Cosine	405.8308		Cosine	609.6965
	Cityblock	410.8807		Cityblock	487.6149
	Direction-aware	**802.3132**		**Direction-aware**	**633.2244**
D1024	Euclidean	368.2901	OP	Euclidean	406.5342
	Cosine	514.7207		Cosine	418.5355
	Cityblock	721.6852		Cityblock	361.4222
	Direction-aware	**838.6839**		**Direction-aware**	**434.6537**
D15	Euclidean	30834.3331	CA	**Euclidean**	81.8571
	Cosine	27464.4951		Cosine	109.6599
	Cityblock	19788.1358		Cityblock	84.0488
	Direction-aware	**36783.2175**		Direction-aware	**115.3565**

As we can see from Table II, in the previous section, the performance of the k-means algorithm is largely influenced by the choice of the type of distance/similarity. Based on the Calinski-Harabasz (CH) indexes of the clustering results, one can see that the k-means algorithm with the proposed direction-aware distance can produce higher quality clusters compared with the one with traditional distances/dissimilarities.

Then, numerical experiments for the same benchmark problems as tabulated in Table I are conducted to evaluate the performance of the evolving algorithm employing the direction-aware distance. To better demonstrate the performance of the evolving algorithm using the direction-aware distance, we involve the following algorithms for comparison:

i) Subtractive clustering algorithm [23];

ii) Mean-shift clustering algorithm [14];

iii) DBScan clustering algorithm [24];

iv) Mode identification based clustering algorithm [25];

v) Random swap algorithm [26];

vi) Density peak algorithm [27].

As the k-means algorithm exhibits certain degree of randomness, we exclude it from the comparison. In the experiments, due to the insufficient *prior* knowledge, we use the recommended settings of the free parameters from the published literature. The experimental setting of the free parameters of the algorithms are presented in Table III.

TABLE III. Experimental Settings of the Algorithms

Algorithm	Free Parameter(s)	Experimental setting
Subtractive	initial cluster radius, r	$r = 0.3$ [23]
Mean-shift	*i)* bandwidth, p *ii)* kernel function type	*i)* r= 0.15 [28] *ii)* Gaussian kernel
DBScan	*i)* cluster radius, r *ii)* minimum number of data samples within the radius, m	*i)* the value of the knee point of the sorted m-dist graph *ii)* m=4 [24]
Mode identification	grid size	Default [25]
Random swap	number of class	number of class [26]
Density peak	*i)* minimum distance, ρ *ii)* local density, δ	*i)* relatively high, ρ *ii)* high, δ [27]

To objectively compare the performance of different algorithms, we consider the following measures:

i) Number of clusters (C), which should be equal or larger than the number of classes in the dataset. However, if C is too large (in our paper, we consider $C > 0.1 \times Number\ of\ Samples$ as too large) or is smaller than the number of classes in the dataset, the clustering result should be considered as an invalid one. The former case indicates that there are too many trivial clusters generated which are hard for users to understand. The latter case implies that the clustering algorithm fails to separate the data samples from different classes.

ii) Calinski Harabasz index (CH) [22], the higher the Calinski Harabasz index is, the better the clustering result is;

iii) Purity (P) [28], which is calculated based on the result and the ground truth:

$$P = \frac{\sum_{i=1}^{N} S_D^i}{K} \tag{21}$$

where S_D^i is the number of data samples with the dominant class label in the i^{th} cluster. The higher purity the clustering result has, the stronger separation ability the clustering algorithm exhibits.

iv) Davies-Bouldin (DB) index [29], the lower Davies-Bouldin index is, the better the clustering result is.

v) Time: the execution time (in seconds) should be as small as possible.

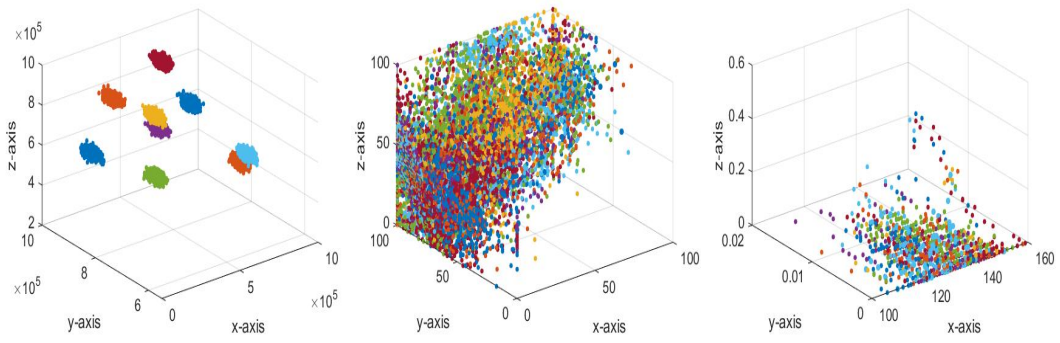
The experiment results obtained by the proposed evolving algorithm as well as other clustering algorithms are given in Table IV. The clustering results of the dim15, Pen-based recognition and Cardiotocography datasets obtained by the proposed algorithm are depicted in Fig. 1, where dots in different colours represent data samples in different clusters.

TABLE IV. Experimental Results

Dataset	Algorithm	C	CH	P	DB	Time	Validity [a]
D256	***The proposed***	16	**203865.1622**	**1.0000**	**0.0248**	1.61	O
	Subtractive	16	**203865.1622**	**1.0000**	**0.0248**	2.86	O
	Mean-shift	103	44374.6685	1.0000	0.3728	**0.19**	O
	DBScan	16	173.1715	0.7598	1.0104	0.21	O
	Mode identification	112	41989.1015	1.0000	0.3736	66.68	×
	Random swap	16	1.0259	0.1221	15.2841	16.03	O
	Density peak	14	597.5327	0.8750	0.6610	1.52	×
D512	***The proposed***	16	**330337.8605**	**1.0000**	**0.0204**	2.15	O
	Subtractive	16	**330337.8605**	**1.0000**	**0.0204**	4.22	O
	Mean-shift	149	56283.7373	1.0000	0.3974	0.52	×
	DBScan	16	203.2336	0.7891	1.0046	**0.32**	O
	Mode identification	1024	NaN	1.0000	0.0000	724.09	×
	Random swap	16	1.1962	0.1260	15.0519	30.76	O
	Density peak	12	291.1243	0.7500	0.8889	1.66	×
D1024	***The proposed***	16	**718469.7967**	**1.0000**	**0.0132**	3.66	O

	Subtractive	16	**718469.7967**	**1.0000**	**0.0132**	11.37	O
	Mean-shift	120	126798.4888	1.0000	0.4496	0.88	×
	DBScan	16	381.3919	0.8721	0.9975	**0.57**	O
	Mode identification	1024	NaN	1.0000	0.0000	2080.58	×
	Random swap	16	0.9093	0.1152	16.3316	71.11	O
	Density peak	14	529.5497	0.8750	0.6965	3.29	×
D15	***The proposed***	9	**302436.3684**	**1.0000**	**0.1177**	13.18	O
	Subtractive	9	**302436.3684**	**1.0000**	**0.1177**	11.28	O
	Mean-shift	9	**302436.3684**	**1.0000**	**0.1177**	**0.04**	O
	DBScan	9	20602.0570	0.9586	1.2317	10.82	O
	Mode identification	3	4327.2420	0.3333	0.5837	141.34	O
	Random swap	9	126.0758	0.2575	10.8063	7.54	O
	Density peak	4	4533.2627	0.4444	0.6696	12.23	×
ST	***The proposed***	23	**2784.0320**	**0.5064**	1.8149	1.62	O
	Subtractive	4	494.1967	0.3988	0.9100	0.66	×
	Mean-shift	1555	24.7451	0.9948	9.8535	2.92	×
	DBScan	18	57.8279	0.48583	1.7112	0.42	O
	Mode identification	9	690.3357	0.3653	**0.3034**	69.05	O
	Random swap	5	1.6124	0.4086	13.2612	2.15	O
	Density peak	3	1224.2338	0.3478	0.4226	2.40	×
PE	***The proposed***	161	**572.8011**	**0.9446**	**1.3937**	10.09	O
	Subtractive	187	382.6055	0.8454	1.9995	12.38	O
	Mean-shift	8501	154.0923	0.9999	0.3652	169.14	×
	DBScan	38	312.9177	0.6209	1.4997	14.04	O
	Mode identification	4316	46.6194	0.9968	0.4969	4243.31	×
	Random swap	10	1.1696	0.1160	77.2047	**9.24**	O
	Density peak	7	2559.6071	0.5993	1.3044	12.65	×
OP	***The proposed***	139	**80.4085**	**0.9247**	**2.0033**	17.46	O
	Subtractive	5620	NaN	1.0000	0.0000	42.07	×
	Mean-shift		No result after 10 hours				×
	DBScan	5	80.5137	0.2190	5.5459	3.88	×
	Mode identification	5620	NaN	1.0000	0.0000	27368.18	×
	Random swap	10	1.7029	0.1142	31.2458	**14.35**	O
	Density peak	8	71.5796	0.2962	1.4627	6.16	×
CA	***The proposed***	113	**231.0072**	**0.8758**	1.0824	1.93	O
	Subtractive	254	140.7584	0.9147	1.3239	0.65	×
	Mean-shift	1594	181.2899	0.9962	0.4175	2.91	×
	DBScan	13	35.8486	0.8053	1.5204	**0.43**	O
	Mode identification	328	63.5207	0.9008	0.6740	40.26	×
	Random swap	3	47.2156	0.7785	5.2548	1.42	O
	Density peak	3	63.5735	0.7813	**0.5081**	2.71	O

[a] "×" stands for invalid results, "O" stands for valid result



(a) dim15 (b) Pen-based recognition (c) Cardiotocography

Fig.1. Visualization of clustering results

From Table IV one can see that the subtractive clustering algorithm is able to produce high quality clustering results on the datasets with Gaussian distribution. However, for the more complex benchmark datasets, it fails to give valid results. The mean-shift clustering algorithm is one of the most efficient algorithms, but it can only perform high-quality clustering with low dimensional datasets. The DBScan algorithm is very efficient as well, but the quality of its clustering results is very limited in terms of the 3 clustering quality measures. Mode identification based clustering algorithm is a so-called "non-parametric" clustering algorithm. Nonetheless, its performance is very limited on high dimensional problems; its computational efficiency is also not very good. The quality of the clustering results obtained by the random swap algorithm is also very limited. In addition, this algorithm requires the number of classes to be known in advance in order to perform valid clustering results; its computational efficiency is also relatively lower. The density peak clustering algorithm is highly efficient, however, based on the recommended input selection, the algorithm failed to separate data samples from different classes in many cases. In addition, with the growth of the number of data samples, the difficulty of deciding the input selection for the users is also increasing.

In contrast, the proposed evolving clustering algorithm consistently produces the top quality clustering results on various problems. Its computational efficiency does not deteriorate with the increase of dimensionality. Therefore, one can conclude that the proposed evolving clustering algorithm is the top one in the comparison.

6. Conclusion

In this paper, a new type of distance, named "direction-aware", is proposed and proved to be a full metric. The proposed distance is defined as a combination of two components: *i)* the traditional Euclidean distance and *ii)* a cosine similarity based angular/directional divergence. Therefore, it is able to consider both spatial and angular divergences. It is using the advantages of one of them to compensate for the disadvantages of the other. The proposed distance is applicable to various traditional machine learning algorithms as an alternative distance measure. A new direction-aware distance based evolving clustering algorithm is also proposed for streaming data processing. Numerical examples demonstrate that the proposed distance can improve the clustering quality of the k-means algorithm for high dimensional problems. They also show the validity and effectiveness of the proposed evolving algorithm for handling high dimensional streaming data.

As future work, we will apply the proposed distance to various high dimensional problems including, but not limited to, the NLP, image processing problems, etc.

References

[1] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," *5th Berkeley Symp. Math. Stat. Probab. 1967*, vol. 1, no. 233, pp. 281–297, 1967.

[2] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Inf. Theory*, vol. 21, no. 1, pp. 32–40, 1975.

[3] G. J. McLachlan, "Mahalanobis distance," *Resonance*, vol. 4, no. 6, pp. 20–26, 1999.

[4] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[5] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International Conference on Database Theory*, 2001, pp. 420–434.

[6] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is 'nearest neighbors' meaningful?," in *International Conference on Database Theoryheory*, 1999, pp. 217–235.

[7] F. A. Allah, W. I. Grosky, and D. Aboutajdine, "Document clustering based on diffusion maps and a comparison of the k-means performances in various spaces," in *IEEE Symposium on Computers and Communications,* 2008, pp. 579–584.

[8] N. Dehak, R. Dehak, J. Glass, D. Reynolds, and P. Kenny, "Cosine Similarity Scoring without Score Normalization Techniques," *Proc. Odyssey 2010 - Speak. Lang. Recognit. Work. (Odyssey 2010)*, pp. 71–75, 2010.

[9] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front end factor analysis for speaker verification," *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[10] V. Setlur and M. C. Stone, "A linguistic approach to categorical color assignment for data visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 698–707, 2016.

[11] M. Senoussaoui, P. Kenny, P. Dumouchel, and T. Stafylakis, "Efficient iterative mean shift based cosine dissimilarity for multi-recording speaker clustering," *IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 7712–7715, 2013.

[12] D. K. Callebaut, "Generalization of the Cauchy-Schwarz inequality," *J. Math. Anal. Appl.*, vol. 12, no. 3, pp. 491–494, 1965.

[13] B. McCune, J. B. Grace, and D. L. Urban, *Analysis of Ecological Communities*. 2002.

[14] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.
[15] J. M. Keller and M. R. Gray, "A fuzzy k-nearest neighbor algorithm," *IEEE Trans. Syst. Man Cybern.*, vol. 15, no. 4, pp. 580–585, 1985.
[16] P. Angelov, X. Gu, and D. Kangin, "Empirical data analytics," *Int. J. Intell. Syst.*, DOI 10.1002/int.21899, 2017.
[17] "Clustering datasets," *http://cs.joensuu.fi/sipu/datasets/*.
[18] "Steel Plates Faults Dataset," *https://archive.ics.uci.edu/ml/datasets/Steel+Plates+Faults*.
[19] "Pen-Based Recognition of Handwritten Digits Dataset," *http://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits*.
[20] "Optical Recognition of Handwritten Digits Dataset," *https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits*.
[21] "Cardiotocography Dataset," *https://archive.ics.uci.edu/ml/datasets/Cardiotocography*.
[22] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Stat. Methods*, vol. 3, no. 1, pp. 1–27, 1974.
[23] S. L. Chiu, "Fuzzy model identification based on cluster estimation.," *Journal of intelligent and Fuzzy systems*, vol. 2, no. 3. pp. 267–278, 1994.
[24] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *International Conference on Knowledge Discovery and Data Mining*, 1996, vol. 96, pp. 226–231.
[25] J. Li, S. Ray, and B. G. Lindsay, "A nonparametric statistical approach to clustering via mode identification," *J. Mach. Learn. Res.*, vol. 8, no. 8, pp. 1687–1723, 2007.
[26] P. Franti, O. Virmajoki, and V. Hautamaki, "Probabilistic clustering by random swap algorithm," in *IEEE International Conference on Pattern Recognition*, 2008, pp. 1–4.
[27] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1493–1496, 2014.
[28] R. Dutta Baruah and P. Angelov, "Evolving local means method for clustering of streaming data," *IEEE Int. Conf. Fuzzy Syst.*, pp. 10–15, 2012.
[29] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 2, pp. 224–227, 1979.