Artificial Intelligence Review
https://doi.org/10.1007/s10462-020-09840-7

Analytical review of clustering techniques and proximity measures

Vivek Mehta[1] · **Seema Bawa**[1] · **Jasmeet Singh**[1]

Abstract
One of the most fundamental approaches to learn and understand from any type of data is by organizing it into meaningful groups (or clusters) and then analyzing them, which is a process known as cluster analysis. During this process of grouping, proximity measures play a significant role in deciding the similarity level of two objects. Moreover, before applying any learning algorithm on a dataset, different aspects related to preprocessing such as dealing with the sparsity of data, leveraging the correlation among features and normalizing the scales of different features are required to be considered. In this study, various proximity measures have been discussed and analyzed from the aforementioned aspects. In addition, a theoretical procedure for selecting a proximity measure for clustering purpose is proposed. This procedure can also be used in the process of designing a new proximity measure. Second, clustering algorithms of different categories have been overviewed and experimentally compared for various datasets of different domains. The datasets have been chosen in such a way that they range from a very low number of dimensions to a very high number of dimensions. Finally, the effect of using different proximity measures is analyzed in partitional and hierarchical clustering techniques based on experiments.

Keywords Unsupervised learning · Hierarchical clustering · Partitional clustering · Proximity measures

1 Introduction

Clustering is one of the most essential techniques applied across a wide range of domains such as in image segmentation, text mining, market research and finance. This technique segregates a collection of data points into separate groups (clusters) for "maximizing intraclass similarity and minimizing interclass similarity" (Han et al. 2011). Thus, all the similar points are grouped into a cluster and the clusters themselves are dissimilar to each other. This partitioning process is performed using a certain proximity measure, density measure and other similar measures. Unlike the process of classification which

✉ Vivek Mehta
vivekmehta27@gmail.com

[1] Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, Punjab 147001, India

Published online: 02 May 2020

requires labels for data points, clustering does not require the knowledge of labels to recognize patterns in a given dataset. This is considerably significant, because in many situations it may either be tedious or expensive to gather the labeling information for a dataset (such as in the case of images and web documents etc.). The broad categories of clustering methods are as follows: hierarchical, partitional, density-based, grid-based and model-based. K-means is a widely used partitional clustering algorithm in which the sum of squares of distances between the center of a cluster and data points is minimized to obtain an optimal data partition of a given dataset. Each data point in this partitioning belongs exactly to one cluster.

However, assigning a data point to exactly one cluster becomes inadequate when there are some data points in a dataset that are almost at an equal distance from two or more clusters. In this case, k-means forcefully assigns them to a single cluster, although they could relate to multiple clusters with the same or varying membership degrees. Thus, this observation wherein data points in a dataset could belong to multiple clusters led to the foundation of fuzzy cluster analysis (Bezdek 1981) where data points are assigned membership degrees to which they belong to a particular cluster. This degree of membership is denoted by μ and lies in the range (0,1). Thus, this type of assignment captures the cluster structure in a more natural way than the traditional hard (crisp) assignments of the k-means algorithm especially in cases where clusters overlap.The evolution of Fuzzy C-Means (FCM) clustering with respect to proximity measures is also reviewed in this paper.

At the heart of every partitional and hierarchical clustering algorithm, lies a proximity measure that indicates how similar or dissimilar two data points are with one another. Some common examples of proximity measures are Euclidean distance, Minkowski distance, Chebyshev distance and Manhattan distance. In a previous study (Lin et al. 2014), it was emphasized that the efficiency of a particular proximity measure used for clustering depends on three factors namely: the *clustering algorithm* used, the *domain* to which a clustering algorithm has been applied and the *feature format* used. In addition, a comparison study (Shirkhorshidi et al. 2015) emphasized one more factor namely the *dimensionality* of a dataset. Considering these crucial aspects of data clustering, this paper presents a review and analysis of many important proximity measures. Based on this review, a procedure is proposed for selecting an appropriate measure for a clustering task. Second, several clustering algorithms are reviewed and their performances are compared on a wide range of datasets of varying dimensions.

Clustering algorithms have been reviewed in many studies. Early important studies (Jain 2010; Jain et al. 1999; Jain and Dubes 1988; Xu and Wunsch 2005) have presented a thorough review of different clustering algorithms; however, proximity measures and data dimensionality were not a prime focus. Among recent studies Xu and Tian (2015), Kameshwaran and Malarvizhi (2014), Fahad et al. (2014), Shirkhorshidi et al. (2014), Cetinkaya et al. (2015) and Sehgal and Garg (2014), some have reviewed clustering algorithms from a big data perspective whereas others have presented a theoretical review of various clustering algorithms highlighting their pros and cons. However, all are very different from the perspective covered in this paper. Specifically for proximity measures, Huang (2008) studied the effect of similarity measures on text document clustering and Strehl et al. (2000) compared the effect of four proximity measures on web page clustering. In a relatively recent research (Shirkhorshidi et al. 2015), many proximity measures were compared based on the performance of clustering algorithms; however, the range of data dimensionality was not too large to cover the data sets of different domains. Considering all these factors, following are the major contributions of this paper.

(1) Effect of preprocessing aspects such as *scales of variables, existence of correlation among features* and *sparse structure of data* on various proximity measures is analyzed. In addition, a theoretical procedure is proposed which can be used to select a proximity measure for clustering. This procedure can also be used while designing a new proximity measure.
(2) Clustering algorithms of different categories are compared experimentally on wisely chosen datasets from different domains. These datasets lie across a wide range of a very low number of dimensions to a very high number of dimensions.
(3) The effect of using different proximity measures is also analyzed in partitional and hierarchical clustering techniques based on experiments.

The organization of this paper is as follows. In Sect. 2, various proximity measures are discussed in detail and their features corresponding to contribution 1 are presented in a tabular form. Section 3 gives an overview of various categories of clustering algorithms. Their advantages and disadvantages are also presented comprehensively. In addition, the evolution of fuzzy clustering with respect to different proximity measures is presented. Section 4 presents the theoretical procedure of key contribution 1. Section 5 provides details regarding all experiments conducted for contributions 2 and 3.

2 Proximity measures for numeric data

A dataset is a collection of data objects and a data object represents an entity that can be, for example, a patient in a medical database, a document in a document dataset, a student in a university database, and an image in an image dataset. Each data object is represented by some attributes that are also known as features. In data analysis tasks such as clustering, classification and outlier analysis, the first and very crucial step is to calculate the proximity (similarity/dissimilarity) between two data objects. If important factors such as sparsity in data, correlation among data features and feature format are ignored in this step, meaningful patterns may remain obscure (McCune et al. 2002). The first and obvious distinction to be made for selecting a proximity measure is the category (or type) to which an attribute belongs. These four basic categories are nominal, binary, ordinal and numeric (Han et al. 2011) as shown in Fig. 1. Nominal attributes contain names that represent a category; hence, these are also known as categorical attributes. There is no specific order or sequence to be followed for these names. In the case where nominal attributes contain only two values, attributes are known as binary attributes. Ordinal attributes are same as nominal but a specific order exists among the labels/names. Lastly, numeric attributes consists of numerical values.

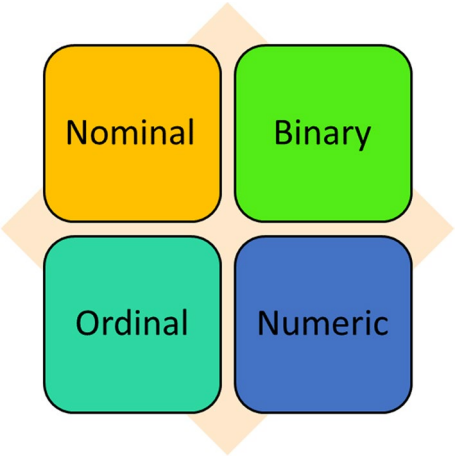
This paper specifically focuses on proximity measures that deal with numeric type of data attributes. These are discussed in the next subsection.

2.1 Popular proximity measures

(1) *Euclidean distance* Let $p_i = (p_{i1}, p_{i2}, \ldots, p_{in})$ and $p_j = (p_{j1}, p_{j2}, \ldots, p_{jn})$ be two data points in some dataset X. The Euclidean distance between p_i and p_j is given as

$$d^2(p_i, p_j) = (p_{i1} - p_{j1})^2 + (p_{i2} - p_{j2})^2 + \cdots + (p_{in} - p_{jn})^2 \quad (1)$$

Fig. 1 Types of attributes



(2) *Manhattan distance (or city block distance)* This distance is defined as

$$d(p_i, p_j) = |p_{i2} - p_{j2}| + |p_{i2} - p_{j2}| + +|p_{in} - p_{jn}| \tag{2}$$

That is, it walks along only one of the axis at a time. This is analogous to walking in a city of blocks where one cannot go diagonally between two locations; instead, we have to walk along either of the two dimensions at a time.

(3) *Minkowski distance* This distance measure is given by the function:

$$d(p_i, p_j) = \sum_{k=1}^{n} (|p_{i,k} - p_{j,k}|)^x)^{1/x} \tag{3}$$

Note that, for $x = 1$, it becomes Manhattan distance and for $x = 2$, it becomes Euclidean distance. Hence, it is a general form of both of them.

(4) *Chebyshev distance* For two vectors, this metric is defined as the maximum of the difference across any of the dimensions of vectors; hence it is also known as the maximum metric. Thus, for two data points $p_i = (p_{i1}, p_{i2}, \ldots, p_{in})$ and $p_j = (p_{j1}, p_{j2}, \ldots, p_{jn})$, Chebyshev distance is calculated as

$$d(p_i, p_j) = \max_n(|p_{in} - p_{jn}|) \tag{4}$$

(5) *Cosine similarity* For sparse datasets (those in which a significant number of zeros are present), the aforementioned traditional measures often do not work well. For example, in document clustering, the representation of a document often consists of a large number of zeros, making the dataset sparse. In such cases, cosine similarity is often used for measuring the similarity between two documents (Han et al. 2011). It is calculated as the cosine value of angle between vectors that represent two documents.

$$sim(d_a, d_b) = \frac{d_a.d_b}{||d_a||.||d_b||} \tag{5}$$

Here, $d_a.d_b$ is the dot product between the vectors d_a and d_b. $||d_a||$ and $||d_b||$ denote the length of the vectors d_a and d_b respectively.

(6) *Pearson distance* Based on the Pearson correlation, the Pearson distance is defined as

$$1 - Corr(a, b) \tag{6}$$

Here, $Corr(a, b)$ is the Pearson correlation of two variables a and b. It is defined as

$$Corr(a, b) = \frac{Cov(a.b)}{\sigma_a.\sigma_b} \tag{7}$$

where $Cov(p, q)$ is the covariance between a and b. σ_a and σ_b are the standard deviations of a and b respectively.

The Pearson distance lies in [0, 2]. This measure has been shown as sensitive to outliers (Hanna et al. 2010) by Anscombe who highlights the importance of studying graphs.

(7) *Kullback–Leibler Divergence (KLD)* If a dataset is assumed to be following some probability distribution, then a measure proposed by Kullback and Leibler (1951) calculates the distance between two probability distributions P_a and P_b as

$$D_{KL}(P_a||P_b) = \sum_i P_a(i) log \frac{P_a(i)}{P_b(i)} \tag{8}$$

This measure lies in the category of non-metric because it is not a symmetric measure and secondly it does not satisfy the triangle inequality (Glen 2018b). Huang (2008) compared the effectiveness of the average KLD measure with other measures such as Euclidean, Cosine, Jaccard and Pearson in the domain of text document clustering.

(8) *Canberra distance metric* This distance metric is used when data vectors contain all non-negative elements (Schoenharl and Madey 2008). For the n-dimensional vectors p and q, it is formulated as

$$d(p_i, q_i) = \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i + q_i|} \tag{9}$$

(9) *Bray–Curtis* This dissimilarity measure is specifically used in the field of ecology and biology to calculate the difference between the counts of species existing on two different sites. It is formulated as

$$BC_{ab} = 1 - \frac{2C_{ab}}{S_a + S_b} \tag{10}$$

where S_a and S_b are the counts at two sites a and b, and C_{ab} is the sum total of smaller counts for each species on both the sites (Glen 2018a).

(10) *Jaccard similarity coefficient* Between two finite sample sets, it measures the similarity as the ratio of the intersection and the union of two sets (say S_1 and S_2) [1].

$$J(S_1, S_2) = \frac{|S_1 \bigcap S_2|}{|S_1 \bigcup S_2|} \tag{11}$$

In generalized form, the Jaccard index between two vectors $P = (p_1, p_2, p_3, \ldots, p_n)$ and $Q = (q_1, q_2, q_3, \ldots, q_n)$ is calculated as

Table 1 Summary for the data handling potential of proximity measures

Proximity measures	Capability factors			
	Metric	Affected by variable's scale	Leverages sparsity	Leverages correlation
Euclidean distance	Yes	Yes	No	No
Manhattan distance	Yes	Yes	No	No
Minkowski distance	No	No	No	No
Chebyshev distance	Yes	Yes	No	No
Cosine similarity	No	No	Yes	No
Pearson correlation	No	No	No	Yes
KLD	No	Yes	Yes	No
Canberra distance metric	Yes	No	No	No
Bray–Curtis	No	No[a]	No	No
Jaccard coefficient	Yes	Yes	No	No
Dice coefficient	No	No	Yes	No
Mahalanobis distance:	Yes	No	No	Yes

[a]Only if the scales for all the variables are same

$$J(P,Q) = \frac{\sum_i min(p_i, q_i)}{\sum_i max(p_j, q_j)} \tag{12}$$

(11) *Dice coefficient* For two vectors x and y, the dice coefficient (Lin et al. 2014) is given as

$$S_{Dic}(x,y) = \frac{2x.y}{x.x + y.y} \tag{13}$$

(12) *Mahalanobis distance* For two vectors u and v, Mahalanobis distance (Shirkhorshidi et al. 2015) is given by

$$d_{mah} = \sqrt{(u-v)S^{-1}(u-v)^T} \tag{14}$$

Here, S is the covariance matrix of the dataset.

All the aforementioned proximity measures have been mathematically analyzed in this paper with respect to some preprocessing aspects. These are the effects of scales of variables, sparse structure of data, existing correlation between the attributes, and metric/non-metric. This analysis is presented in Table 1.

2.2 Some domain specific proximity measures

Apart from proximity measures mentioned in the previous subsection, it is worth mentioning some proximity measures that are relatively recent and have been designed in pertinent to a specific domain such as text mining and image analysis.

2.2.1 Proximity measures in text mining

(1) *Extensive Similarity (ES)* This similarity measure (Basu and Murthy 2015), extensively takes each and every document d_k present in the corpus to determine the similarity between the two documents d_i and d_j. According to ES, two documents are said to be exactly similar to each other if they both are similar to each other and they both are either similar or dissimilar to every other document contained in the corpus.

The first step of ES is to calculate the value of $dis(di, dj)$ as follows.

$$dis(d_i, d_j) = \begin{cases} 1, & \text{if } \rho(d_i, d_j) \leq \Theta \\ 0, & \text{otherwise .} \end{cases} \tag{15}$$

where $\rho(d_i, d_j)$ is a similarity measure (cosine has been used in the original work) and $\theta \in (0, 1)$. If $dis(d_i, d_j) = 0$, a score l is assigned as follows.

$$l_{i,j} = \sum_{k=1}^{N} |dis(d_i, d_k) - dis(d_j, d_k)| \tag{16}$$

Here, N denotes the total documents in the corpus, and finally, ES for two documents d_i, d_j is

$$ES(d_i, d_j) = \begin{cases} N - l_{i,j}, & \text{if } dis(d_i, d_j) = 0 \\ -1, & \text{otherwise .} \end{cases} \tag{17}$$

Thus, two documents d_i and d_j can have a maximum ES value of N when the distance between them is zero and for every k, the distance between d_i and d_k and the distance between d_j and d_k is the same.

(2) *Similarity Measure for Text Processing (SMTP)* This measure considers the absence and presence of a feature in two documents to be more significant than the difference of feature values. For example, if a feature w_1 is absent in d_1 but present in d_2 so that $d_{11} = 0$ and $d_{21} = 2$, and another feature w_2 is present in both d_1 and d_2 so that $d_{12} = 3$ and $d_{22} = 5$, then w_1 is considered to be more important than w_2 in calculating the similarity between the documents d_1 and d_2 despite of the same difference value which is 2. This property was shown to remain unsatisfied by other traditional proximity measures such as Euclidean, Cosine, Dice coefficient and IT-Sim etc in a previous study (Lin et al. 2014). Additionally, the study indicated that the usefulness of a similarity measure strongly depends on the following factors:

(a) Applications domains (e.g., image or text).
(b) Representation format of the feature, for example, Term Frequency Inverse Document Frequency (Tf-IDF) or word count in case of a text document.
(c) The classification/clustering algorithms used.

These results form the basis to propose a theoretical procedure in Sect. 4.

(3) *DRSim* Between the two document vectors x_i and x_j, DRSim (Saraçoğlu et al. 2007) finds the similarity as

$$DRSIM(x_i, x_j) = \left(\frac{\sum_{k=1}^{m} |x_{ik} - x_{ij}|^2}{m} \right)^{1/m} \tag{18}$$

where m denotes the dimension of the feature vectors of documents. The results were shown to be better than those achieved using the Minkowski distance measure (Saraçoğlu et al. 2007).

(4) *Style based* In a previous study (Leoncini et al. 2011), a novel similarity metric was proposed that considers the position of concepts(terms) in a document. The idea behind this was that two similar documents should share some structural arrangement of terms contained in them. Thus, the final metric of this study contained two aspects, one for concepts of terms derived using EuroWordNet (Vossen 2002) ontology and the other as position of concepts.

(5) *Kernel induced* To capture patterns contained in the form of non-spherical clusters, Kannan et al. (2012) used kernel functions instead of Euclidean distance. Kernel functions map the original feature space to a higher dimension space by using some non-linear transformation (such as Gaussian kernel, sigmoid kernel and polynomial kernel). The new distance function obtained out of this is as follows:

$$d^2(x_k - v_i) = 2(1 - K(x_k, v_i)) \tag{19}$$

where

$$K(x, y) = exp\Big(\frac{-||x - ||^2}{2\sigma^2}\Big) \tag{20}$$

3 Clustering algorithms

Proximity measures are a crucial part of most clustering algorithms. Having reviewed them in the previous section, in this section, a description of various clustering algorithm categories is provided.

3.1 Classification of clustering algorithms

Figure 2 shows a broad classification of clustering algorithms (Han et al. 2011). A brief description of various categories is as follows:

(I) *Partitioning based* Let a dataset D contains n number of objects. Given a value k (where $k \leq n$), partitioning methods partition the n objects into k clusters $C_1, C_2, \ldots, C_k$. The following conditions are to be satisfied by the obtained partitions: (1) none of the clusters should be empty and (2) each object must be contained in either one (Hard c-means) or more than one (Fuzzy c-means) clusters. First, k cluster centers are chosen either randomly or by using some more sophisticated methods and then a relocation method is used to shift the cluster centers towards an optimal solution, for instance, (1) in K-means (MacQueen et al. 1967), the average value of all data points in the cluster is used to find the new cluster center whereas (2) k-medoids (Park and Jun 2009) represent a cluster by an object that is located near to the center of the cluster. The quality of clustering is measured by an objective function. This objective function is designed to achieve high intracluster similarity and low intercluster similarity. Other well-known algorithms in this category are: K-modes (Huang 1997), PAM (Ng and Han 1994),

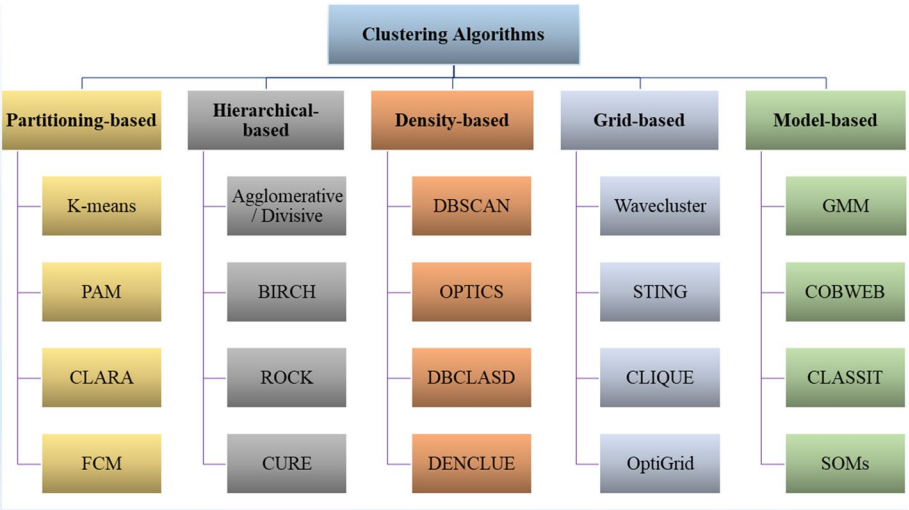


Fig. 2 Broad classification of clustering algorithms

CLARA (Ng and Han 2002) and FCM (Bezdek et al. 1984). A detailed explanation of the K-means technique is described as follows:

K-means clustering algorithm

(1) A set of k points from the dataset D are chosen randomly as centers representing k clusters.
(2) Each point is assigned to the cluster whose center is at the minimum distance from it using Euclidean distance.
(3) Cluster centers are recomputed using current cluster memberships.
(4) Go to step (2) till the membership to clusters stops changing.

Several variants of K-means can be found in the literature (Jain et al. 1999). Some of these target to initialize cluster centers in a more efficient way so as to reach a global optimum, whereas others use a totally different objective function. However, K-means and other similar algorithms of this type tend to get trapped in local optimal solutions. This limitation was overcame when in 1990, the data clustering problem was solved by the application of nature-inspired metaheuristic algorithms by using simulated annealing (Selim and Alsultan 1991). A similar recent approach using the gravitational search algorithm (GSA) is presented in Han et al. (2017). A detailed survey of the usage of these algorithms for data clustering is presented in Nanda and Panda (2014). In recent research, deep neural networks have been used to achieve a non-linear mapping of the original feature space in order to solve the problem of high dimensionality (Xie et al. 2016). Learning of the parameters of the deep neural network and cluster centers was performed simultaneously by optimizing a KLD based objective function. However, because of simplicity and less computational cost, k-means is still used widely.

(II) *Hierarchical based* These methods perform a hierarchical breakdown of a given dataset which can be classified as agglomerative and divisive. In agglomerative methods, initially, each object is regarded as a cluster on its own and they are then successively

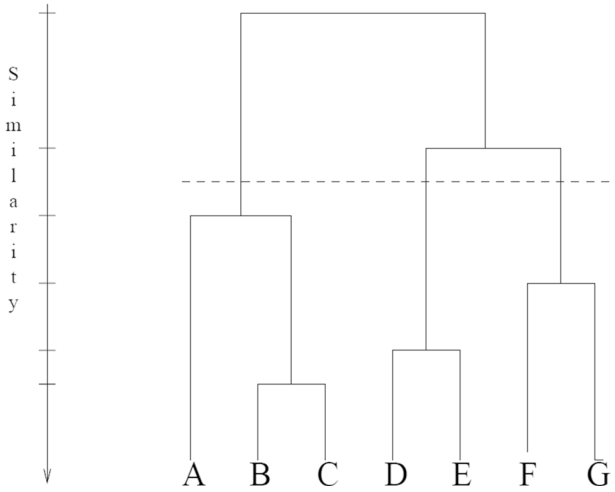


Fig. 3 A dendogram showing clustering hierarchy on 7 points (Jain and Dubes 1988)

merged till they satisfy a termination condition. By contrast, in the divisive approach, initially, the set of objects is considered as a single large cluster and is successively split up into smaller clusters until a termination condition is satisfied. The former is also called the bottom-up approach whereas the latter is called the top-down approach. A general algorithm for agglomerative clustering is as follows.

Agglomerative clustering algorithm

(1) Let each data point be a cluster on its own.
(2) Compute the proximity matrix of individual points.
(3) Merge the two closest clusters and then update the proximity matrix.
(4) Repeat step (3) until a single cluster remains.

In the form of output, a hierarchical clustering algorithm yields a tree-like structure known as a *dendogram* which can be broken at different levels to give corresponding different data clusterings. Depending on how the inter-cluster similarity is defined, three important agglomerative hierarchical clustering algorithms include *single-linkage, complete linkage* and *average linkage*. Single linkage algorithm uses the distance between the *closest* pair of data points in clusters as a measure of inter-cluster similarity. Complete linkage algorithm uses the distance between the *farthest* pair of data points as the inter-cluster similarity; while the average linkage algorithm uses the distance between the *group average* of all data points contained in a cluster as the proximity measure between clusters. Figure 3 (Jain and Dubes 1988) shows a dendogram created as a result of single linkage clustering applied on seven data points. Other popular hierarchical algorithms include BIRCH (Zhang et al. 1996), ROCK (Guha et al. 1998), CURE (Guha et al. 2000) and Chameleon (Karypis et al. 1999).

(III) *Density based* Methods described above find the clusters based on a proximity measure and hence face difficulty while finding clusters of arbitrary shape (Han et al. 2011). On the other hand, density-based methods discover clusters based on density. These methods can find clusters of arbitrary shapes. Here, a cluster is kept growing as long as the number of data objects in the neighborhood exceeds some threshold value. In any

density-based clustering method, *density* at some point p is defined to be the number of data points lying in a circle of radius *eps* around p. Also, if a circle of radius *eps* consists of some minimum number of data points denoted by *minpts* then the region is called the dense region. A *core point* is a point that consists of a dense region around it. A border point is a point that has points less than *minpts* around it but itself lies inside the neighborhood of a core point. Lastly, a point that is neither a core point nor a border point is known as a noise point. DBSCAN (Ester et al. 1996), a basic density-based algorithm can be abstracted as follows (Schubert et al. 2017).

DBSCAN algorithm

(1) Identify all core points.
(2) Assign neighboring core points into a single cluster.
(3) *For each* non-core point *do*
If possible, assign it as a border point to the cluster of the closest core point otherwise, add it to noise.

Other algorithms of this category include OPTICS (Ankerst et al. 1999), DBCLASD (Xu et al. 1998) and DENCLUE (Hinneburg et al. 1998).

(IV) *Grid based* Here, each dimension is divided into several cells thus forming a grid structure between dimensions. Clustering operations are then performed on this quantized space. The processing time of these methods is independent of the number of objects. Rather, it is determined by the number of cells in the grid structure. STING (Wang et al. 1997) , Wavecluster (Sheikholeslami et al. 1998), CLIQUE (Jain and Dubes 1988) and OptiGrid (Hinneburg and Keim 1999) are well-known examples of this category.

An overview of CLIQUE which is one of the initial algorithms in this category is presented as follows (Han et al. 2011).

(1) Partition the d-dimensional data space into non-overlapping rectangular units (or cells) and identify dense units in all subspaces based on a density threshold l.
(2) Dense cells in each subspace are then used to generate clusters by starting with an arbitrary dense cell and finding the maximal region covering the cell and working on the remaining dense cells.

A more detailed description of this algorithm can be found in Agrawal et al. (2005).

(3) *Model based* These methods perform clustering by first hypothesizing a mathematical model and then finding its best fit for a given dataset. For example, the EM algorithm performs an expectation-maximization analysis (Dempster et al. 1977), COBWEB performs a probability analysis (Fisher 1987) and a neural network based method, Self-Organizing Maps (SOMs) (Kohonen 1998), performs clustering by mapping high dimensional data onto a 2-D or 3-D feature map. CLASSIT (Gennari et al. 1989) is an extension algorithm of COBWEB for continuous-valued data.

The advantages and disadvantages of aforementioned clustering algorithms determined from the literature are summarized in Table 2. In Sehgal and Garg (2014), a performance comparison of these algorithms is presented according to the size of datasets and time taken for cluster formation. In Shirkhorshidi et al. (2015), a performance

Table 2 Comparison of different clustering techniques

Clustering technique	Advantages	Disadvantages
Partitioning based	Algorithms converge faster and are robust to noise	Tends to produce only convex clusters. Difficult to work with nominal/ordinal attributes
Hierarchical based	Can find nonconvex clusters. The number of clusters is not required	Fails in the presence of noise and is more time-consuming. It requires large memory space for large datasets
Density-based	Arbitrary shaped clusters can be discovered and the number of clusters is also not required. Has the ability to treat outliers as noise robustly	Clustering quality highly depends on an appropriate selection of parameters. In the case of varying density, algorithms do not perform well (in the case of DBSCAN). It also suffers from the curse of dimensionality
Grid-based	More suitable to high dimensional datasets. The order of input of records has no effect	Clustering quality is highly dependent on the size and number of grid cells
Model-based	There exists a choice for appropriate statistical models to capture latent clusters. Data points are not explicitly assigned instead they have a probability of belongingness to multiple clusters	EM algorithm can be considerably expensive if there are a large number of distributions and the algorithm does not guarantee to get settle on a global optimum (a more serious concern in high dimensions.)

comparison of partitioning algorithms such as K-means and K-medoids, based on different proximity measures was performed; however, the range of data dimensionality did not cover datasets of much larger dimensionality (such as textual data). In this study, experiments to study and compare the performance of various clustering algorithms of different categories are presented (Sect. 5) that covers a considerably wide range of data dimensionality. Having reviewed proximity measures in Sect. 2 and clustering algorithms in Sect. 3.1, in the next subsection, partitioning based algorithm FCM is reviewed especially for proximity measures.

3.2 Evolution of fuzzy clustering based on proximity measures

When it comes to applications where a data object can belong to more than one category (i.e., when a degree of ambiguity or uncertainty is involved), the role of fuzzy cluster analysis comes into the picture to provide a better partitioning of data objects. Fuzzy cluster analysis allows the degree of membership of an object to a cluster to be measured in the range [0, 1] (denoted by μ). This concept of degree of membership allows greater flexibility to express the belongingness of data objects to multiple clusters (Kruse et al. 2007). Given a dataset $X = \{x_1, x_2 \ldots x_n\}$ these memberships lead to the output of the clustering process to be a fuzzy label vector of degree of memberships to all clusters for each data point x_j. The fuzzy label vector can be represented as $\mu_j = (\mu_{1j}, \mu_{2j}, \ldots, \mu_{cj})^T$. The c x n matrix $U = (\mu_{ij})$ is called a fuzzy partition matrix, where c is the number of clusters and n is the total number of data points.

3.2.1 Fuzzy c-means algorithm

Let $D = \{x_1, \ldots, x_n\}$ be the set of data points and the number of required clusters be c $(1 < c < n)$. To find a fuzzy partition matrix $U = (\mu_{ij})$, Dunn (1973) introduced the FCM algorithm. Bezdek (1981) later improved this algorithm. In FCM, the goal of finding the optimum fuzzy c-partition matrix U is encoded with an objective function J_m (Ross 2005) as

$$J_m(U, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu_{ik})^{m'} (d_{ik})^2 \tag{21}$$

where

$$(d_{ik})^2 = (d(x_k - v_i))^2 = \sum_{j=1}^{m} (x_{kj} - v_{ij})^2 \tag{22}$$

here m is the dimension of the dataset. The weighting parameter m' controls the extent to which the membership is shared between clusters. It ranges in $[1, \infty)$. When $m' = 1$, the membership value μ_{ij} is either 0 or 1. By contrast, when $m' \to \infty$, the value of $J_m \to 0$. In general, the bigger the value of m' is, the more the fuzzier is the partition matrix U.

An optimal partition matrix U corresponds to the minimum value of this objective function J_m, which is a solution to the following equation:

$$J_m^*(U^*, v^*) = min\, J(U, v). \tag{23}$$

under the constraints

$$\sum_{j=1}^{n} \mu_{ij} > 0, \forall i \in \{1, \ldots, c\}, \tag{24}$$

and

$$\sum_{i=1}^{c} \mu_{ij} = 1, \forall j \in \{1, \ldots, n\} \tag{25}$$

An iterative algorithm introduced by Bezdek (1981) popularly known as the FCM algorithm to get a solution to this equation is as follows:

(1) Choose c ($1 < c < n$) and m'. Initialize the partition matrix $U^{(0)}$. As in Ross (2005), here, each step is labelled by r, where $r = 0,1,2,...$
(2) Calculate the c centers $v_i^{(r)}$ as:

$$v_{ij} = \frac{\sum_{k=1}^{n} (\mu_{ik})^{m'} . x_{ki}}{\sum_{k=1}^{n} (\mu_{ik})^{m'}} \tag{26}$$

(3) Update the partition matrix as follows:

$$\mu_{ik}^{(r+1)} = \left[\sum_{j=1}^{c} \left(\frac{d_{ik}^{(r)}}{d_{jk}^{(r)}} \right)^{\frac{2}{m'-1}} \right]^{-1} \tag{27}$$

(4) If $||U^{(r+1)} - U^{(r)}|| \leq \varepsilon_L$, stop; else, go to step 2 and make $r = r + 1$.

The next subsection presents different variants of FCM which are observed in the literature in the past 3 decades.

3.2.2 Variants of FCM based on distance functions

Euclidean distance used in FCM had been replaced many times by some other measure leading to better clustering results. An attempt to cover them comprehensively is given below.

(1) *Gustafson-Kessel algorithm*

Euclidean distance which is originally used in FCM favors clusters that are of spherical shape. In Gustafson and Kessel (1979), it was replaced by Mahalanobis distance, thus the algorithm could find clusters of arbitrary shape. Mahalanobis distance related to a cluster i, is given by the equation

$$d^2(x_j, C_i) = (x_j - C_i)^T \sum_i^{-1} (x_j - C_i) \tag{28}$$

where $\sum_i$ is the covariance matrix of the cluster. Cluster centers and membership degrees are calculated in the same way as in original FCM. The covariance matrix (Rudolf Kruse Christian Döring 2007) is updated as

$$\sum_i = \frac{\sum_i^*}{\sqrt[p]{det(\sum_i^*)}} \tag{29}$$

$$\sum_i^* = \frac{\sum_{j=1}^n \mu_{ij}(x_j - c_i)(x_j - c_i)^T}{\sum_{j=1}^n \mu_{ij}} \tag{30}$$

However, owing to matrix inversions, computational costs are higher for this algorithm in comparison with FCM (Rudolf Kruse Christian Döring 2007).

(2) *Pedrycz-97*

Early semi-supervised fuzzy clustering techniques as in Pedrycz and Waletzky (1997) also used Mahalanobis distance. In Pedrycz and Waletzky (1997), a total of three experiments were performed based on three different datasets. The results of that study showed that when Mahalanobis distance was used, the convergence rate was the highest among all algorithms used.

(3) *Kernel-based clustering*

In Wu et al. (2003), a Fuzzy Kernel C-Means algorithm (FKCM) was proposed to deal with datasets that may consist of non-spherical clusters. In kernel-based clustering, the original feature space is transformed into a high dimensional feature space. The transformation of space is denoted as $\phi : X \rightarrow F$, where X is the original feature space and F is the transformed feature space. The transformed data is denoted by $\phi(x)$. In FKCM, FCM is integrated with a mercer kernel function to capture the non-spherical shape of clusters (such as the annular ring shape). The results presented in Wu et al. (2003) showed that for spherical datasets, FCM and FKCM perform equally well, but for annular ring-shaped datasets, FKCM clusters more effectively.

(4) S^2*KFCM*

In Zhang et al. (2004), using Gaussian kernel, a new Semi Supervised Kernel Fuzzy C-Means (S^2KFCM) algorithm was introduced by Zhang et al. Gaussian kernel is given by the equation

$$K(x, y) = exp\left(\frac{-||x - y||^2}{\sigma^2}\right) \tag{31}$$

Experiments conducted on benchmark datasets indicated that better classification results were obtained using S^2KFCM, than other classical algorithms such as K-NN and SVM (Zhang et al. 2004).

(5) *Bouchachia et al.*

Bouchachia and Pedrycz (2006) investigated the effect of four different distance measures named Euclidean, weighted Euclidean, fully adaptive and kernel-based distance with the same objective function. After performing experiments on three datasets (fully labeled), it was found that the relative performance (high to low) was in the order of usage: fully adaptive(e.g., Mahalanobis distance), weighted Euclidean, kernel-based distance and Euclidean distance. Thus, the use of fully adaptive distance yielded the best results.

(6) *Lai and Garibaldi*

Lai and Garibaldi (2011) compared four algorithms with different objective functions [Pedrycz-97 (Pedrycz and Waletzky 1997), Li-08 (Li et al. 2008), Zhang-04 (Zhang et al. 2004) and Endo-09 (Yasunori et al. 2009)] by using different distance metrics,

namely Euclidean distance, Mahalanobis distance and Gaussian kernel-based distance. They indicated that Pedrycz-97 and Li-08 perform better than others owing to the presence of Mahalanobis distance. Because of the presence of the inverse of covariance matrix in Mahalanobis distance, different scales of variables are normalized, and the correlation between features is also handled. The results also showed that the infinity problem arises when using Euclidean distance with high dimensional datasets.

Based on the literature reviewed in this paper, in the next section (4) a concise view of factors upon which an appropriate selection of a proximity measure depends is presented. Based on the factors reviewed, a procedure for the same has also been proposed.

3.3 Recent applications of clustering

Some important and recent applications of clustering are discussed in detail below.

1. *Social news clustering* In recent years, the emergence of social media as a huge platform for information sharing has led news organizations to extract useful information from it. Some recent attempts for the news clustering task involve modeling of pair of words that co-occur in a corpus of short texts (Xia et al. 2015; Yan et al. 2013). In Jan (2020) a cluster-then-label semi-supervised approach for labeling of tweets as spam/not spam was proposed. Grouping of news written in different languages is another interesting area of research; for instance in Hong et al. (2017), the Chinese–Vietnamese news dataset was used for clustering based on the semantic correlation between two languages.
2. *Document clustering* The task of organizing raw text documents into useful categories when no predefined labels are available, is known as document clustering. Document clustering plays a considerably significant role in fast information filtering, automatic document organization, and topic extraction and comprehension. Various statistical and semantic-based document clustering techniques have been proposed by different authors. Statistical techniques mainly rely on frequency-based bag of words (BOW) model (Manning et al. 1999) and other modified term weighting schemes such as Tf-IDF (Altınçay and Erenel 2010; Lan et al. 2005). On the other side, semantic-based techniques take the benefit of lexical databases such as Wordnet (Sedding and Kazakov 2004; Wei et al. 2015). Apart from the clustering of documents written in English language, multilingual document clustering techniques have also been proposed. Some state of the art techniques in this new research area can be found in Tang et al. (2015) and Hong et al. (2017).
3. *Sentiment analysis* Various e-commerce websites such as Amazon and Flipkart offer their users to express their reviews regarding their products and services. Similarly, social networking sites such as Facebook, YouTube, and Twitter, allow users to express their opinions regarding any social event or political news. “Sentiment analysis is the task of detecting, extracting and classifying opinions, sentiments and attitudes concerning different topics, as expressed in textual input” (Montoyo et al. 2012). This type of analysis helps other users on the same platform to make purchase decisions and service providers to improve the quality of services. Clustering is an important step in the whole process of sentiment analysis for grouping similar sentiments together. Ravi and Ravi (2015) presented a comprehensive survey of different techniques used in sentiment analysis. Recent research in this area has focused on (1) a language-independent approach for sentiment analysis such as the one presented in García-Pablos et al. (2018), and (2) a novel vector space model for concept-level sentiment analysis that allows reasoning

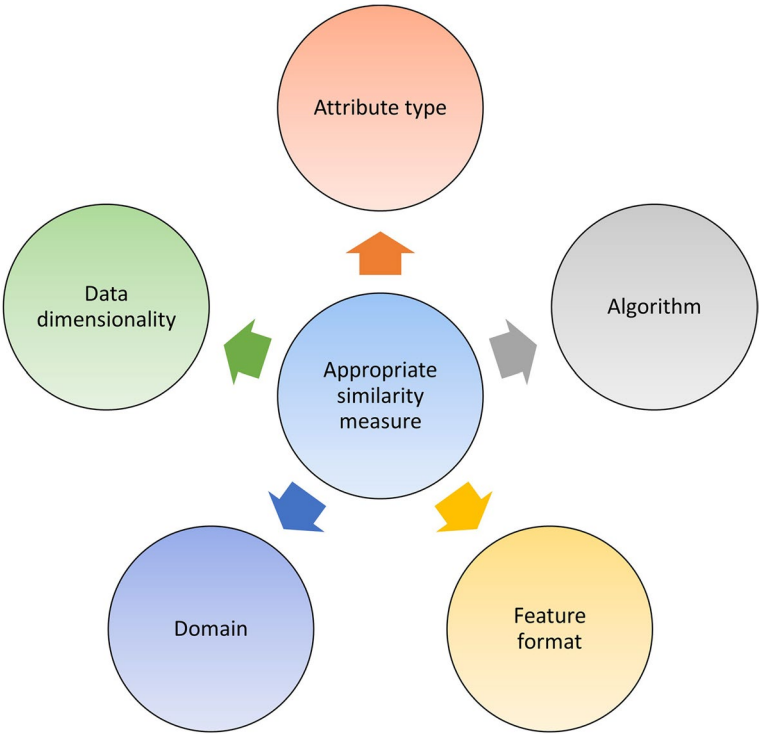


Fig. 4 Factors that determine the usefulness of a proximity measure

by analogy on natural language concepts namely *AffectiveSpace2* (Cambria et al. 2011, 2015).

4 Analysis of the usefulness of a proximity measure

In Sect. 2, many different proximity measures were presented. Various authors have analyzed the usefulness of proximity measures with respect to different factors. These factors are data dimensionality, application domain, feature format and clustering algorithm. This section presents a concise view of all factors that are important in determining the usefulness of a given proximity measure for clustering purposes. Based on the following research papers, different factors that have been identified as important by authors are highlighted in Fig. 4.

In Shirkhorshidi et al. (2015), using high dimensional datasets, clustering using partition-based algorithms was performed and significant results were produced indicating that the average Euclidean distance is the fastest when using the K-means clustering algorithm. In addition, results indicated that Mahalanobis distance is the best performer for low-dimensional datasets.

According to Lin et al. (2014), the efficiency of a particular proximity measure used for clustering depends on three factors: the *clustering algorithm* used, the *domain* to which a clustering algorithm is being applied and the *feature format* used. In addition, a comparison

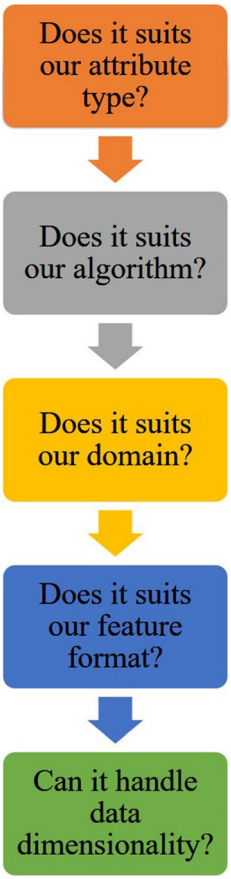


Fig. 5 Proposed procedure for selecting a versatile proximity measure

study (Shirkhorshidi et al. 2015) emphasized one more factor namely *dimensionality* of the dataset. Huang (2008) studied the effect of similarity measures on text document clustering. Strehl et al. (2000) compared the effect of four similarity measures on web page clustering.

Based on this review, a stepwise procedure is proposed in this paper, that can help a researcher to choose and design a versatile proximity measure when the accuracy of clustering results is the primary goal. The procedure is shown in Fig. 5. The principle behind the procedure is as follows.

- *Step1: Selection of the measure based on attribute type*
  It is clear that first, a proximity measure must be chosen depending on the attribute type such as nominal, ordinal, continuous and binary.
- *Step2: Selection of the measure based on the algorithm to be used*
  Depending upon data, (e.g., whether it contains overlapping or non-overlapping clusters) one may use clustering algorithms suitable to our data. As observed in Shirkhorshidi et al. (2015) and also suggested in Lin et al. (2014), different proximity measures perform differently in a clustering algorithm; thus, the second step must be to determine the suitability of the proximity measure to the clustering algorithm.

- *Step3: Selection of the measure based on the domain where clustering is to be applied*
  Each domain has its criteria to be satisfied for a proximity measure to be suitable to it. For example, text clustering has its listing as suggested in Lin et al. (2014), the image segmentation domain has its criteria for a good proximity measure, and the financial domain has its conditions that are to be followed. Thus, the next step to be taken care of while designing or choosing a proximity measure must be the domain knowledge.
- *Step4: Selection of the measure based on the feature format to be used*
  After entering into the domain, the data representation may have one or more forms; for instance, a document can be in the form of BOW model (Manning et al. 1999) or it can be represented in the form of lexical chains (Wei et al. 2015). This will also affect the design of the proximity measure.
- *Step5: Selection of the measure based on data dimensionality*
  Finally, data can be low dimensional or high dimensional. Work performed in Shirkhorshidi et al. (2015) can act as a good reference to choose between a number of measures that can perform better in either of the two situations. Studying the behavior of proximity measures with respect to the dimensionality of a dataset by using partitional and hierarchical clustering algorithms is one of the major experiments in this study as well.

As proposed in Shirkhorshidi et al. (2015), no single proximity measure can be perfect for all types of datasets. It is clear that, although no proximity measure can satisfy all the aforementioned requirements for all types of datasets, however, for a single type of domain (e.g., document clustering or image segmentation etc.), specific measures can be designed out of existing measures. For example, SMTP has been proposed in Lin et al. (2014) which particularly measures the similarity of two documents. The next section presents conducted experiments to compare clustering algorithms of different categories which are presented in Sect. 3. It also details the experiments of studying the effect of using different proximity measures in partitional and hierarchical clustering algorithms on datasets ranging from the low number of dimensions to a very high number of dimensions.

5 Experiments

In this section, an experimental comparison of various clustering techniques of different categories is conducted on various datasets. These datasets are described in the next subsection. Second, experiments of using various proximity measures in partitional and hierarchical clustering algorithms on several datasets with varying dimensionality ranging from low to very high number of dimensions are presented. All these experiments are performed on a machine having Intel(R) Core(TM) i7-6700 processor with 16GB RAM. The source code is written in Python with support from the sklearn machine learning library. Important packages used are sklearn.cluster, scipy.spatial.distance, sklearn.preprocessing and sklearn.metrics.

Table 3 Datasets description

Dataset name	Domain	No. of objects	No. of dimensions	No. of classes
Iris	Medical/biology	150	4	3
Breast cancer	Medical/biology	699	9	2
Glass	Physics	214	9	7
Mfea-Fou	Images, handwritten digits	2000	76	10
Mfea-Fac	Images, handwritten digits	2000	216	10
Mfea-Pix	Images, handwritten digits	2000	240	10
USPS	Images, handwritten digits	9298	256	10
MNIST	Images, handwritten digits	60000	784	10
TOX	Medical/biology	171	5748	4
Leukemia1	Medical/biology	72	7070	3
Wikipedia articles	Text	59701	> 50,000	25
20Newsgroup	Text	> 18,000	> 30,000	20

5.1 Datasets

To setup a good testbed for a comprehensive evaluation, various datasets from different areas, such as image, text and biology are used in experiments. Iris,[1] Breast cancer,[2] leukemia[3] and TOX[1] are taken from the biomedical domain. Mfea-fou, Mfea-fac, and Mfea-pix[4] are taken from a multiple features database consisting of features of handwritten numerals. USPS[5] and MNIST[6] are also scanned images of handwritten digits. The 20Newsgroup (Pedregosa et al. 2011) dataset consists of a large number of textual articles. Lastly, glass,[7] a physics domain dataset consists of attributes defining the type of glass. Table 3 summarizes the details of these publicly available benchmark datasets in the order of increasing number of dimensionality.

5.2 Parameter settings

All clustering algorithms have their own requirements of parameter values to perform the process of clustering. Values used in this paper are stated as follows.

K-means This algorithm does not guarantee a globally optimum solution because it tends to stick in a local bad optimum. For this reason, for each dataset, it has been execued 7 times with different seed values corresponding to which different initial clusters are chosen every time. This algorithm also requires the maximum number of iterations for its convergence which is set to 400 and lastly, the number of clusters (i.e., k value) is set to be the

[1] https://archive.ics.uci.edu/ml/datasets/iris, Accessed: 2019-05-12.
[2] https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/, Accessed: 2019-05-14.
[3] http://featureselection.asu.edu/datasets.php, Accessed: 2019-05-15.
[4] https://archive.ics.uci.edu/ml/datasets/Multiple+Features, Accessed: 2019-05-15.
[5] https://www.openml.org/d/41070, Accessed: 2019-05-16.
[6] https://cs.nyu.edu/~roweis/data.html, Accessed: 2019-05-16.
[7] https://archive.ics.uci.edu/ml/datasets/glass+identification, Accessed: 2019-05-16.

number of available classes for each dataset used in experiments. To investigate the effect of different proximity measures, the algorithm is run with different proximity measures such as Euclidean, correlation distance and Manhattan distance.

Hierarchical agglomerative clustering The first parameter required by this algorithm is the number of clusters which is set to be the number of available classes. Second, the linkage criterion is set to be *average* i.e. to calculate the distance between two clusters, the mean of all data points in a cluster is used.

DBSCAN Although, this algorithm, can find nonconvex clusters, the difficulty lies in determining its two important parameters, namely *eps* value and *minpts* as already defined in Sect. 3.1. To judge *eps* value, a histogram of distances to the nearest neighbor is plotted for each data point in a dataset. Then *eps* value is set to be the distance under which the nearest neighbors for most of the data points lie. To calculate *minpts* value, a histogram is plotted that indicates the count of neighbors for each data point that lies in the *eps* amount of neighborhood. Then, the minimum value from the histogram for which approximately 10% of all data points are contained in its *eps* neighborhood is chosen to be the *minpts* value.

GMM This algorithm requires only the number of components for the Gaussian mixture. This value corresponds to the number of available classes.

5.3 Clustering evaluation metrics

Various metrics are defined in the literature for assessing clustering quality. These are Rand Index (RI), Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), Adjusted Mutual Information (AMI), F-measure, Homogeneity, V-measure, Heterogeneity, Completeness and Silhouette coefficient etc. A few of these metrics that are widely used in the data clustering community are defined as follows.

- *ARI* ARI is used to compare two clustering assignments that ignore permutations and is a chance normalized version of RI. In this study, ARI is used to assess clustering quality. Similar clustering assignments yield a score close to 1.0 whereas non-positive scores are yielded for dissimilar clustering. Let C denotes the ground truth class labeling and K be the clustering assignment. Also, let
  - A be the number of element pairs that lie in the same set of C and K, and
  - B be the number of element pairs that lie in different sets of both C and K.

  Then RI is given by:

$$RI = \frac{A + B}{C_2^{n_{samples}}} \tag{32}$$

  where $C_2^{n_{samples}}$ denotes the total number of possible pairs in the dataset. To overcome the drawback of RI (i.e. random label assignments will not evaluate the RI value close to zero), ARI is defined by discounting the expected RI denoted as $E[RI]$ value of random labelings as follows.

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]} \tag{33}$$

- *AMI* Mutual Information (MI) measures the similarity of two clustering assignments. AMI is a version of MI that is adjusted against chance (Pedregosa et al. 2011). Hence,

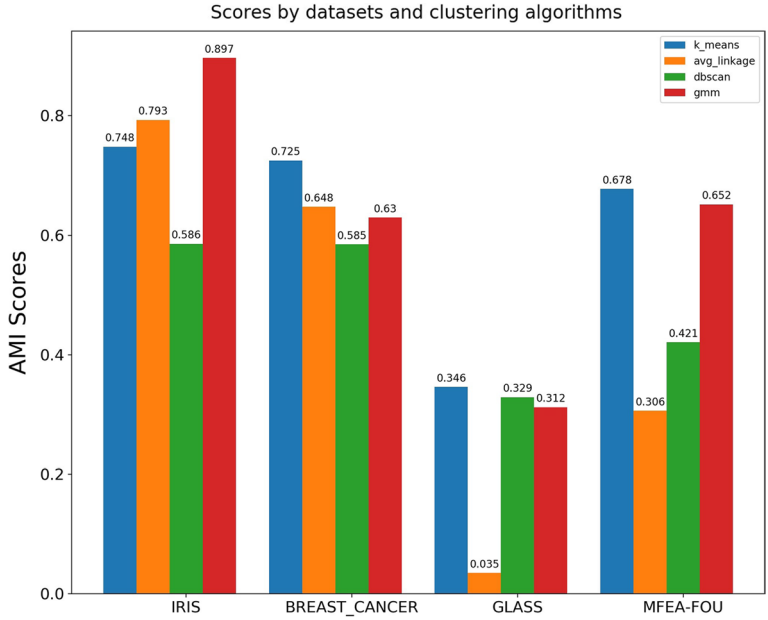


Fig. 6 Clustering results on low dimensional datasets

for datasets for which ground truth labels are available, AMI is used in this study to compare clustering results. Perfect labeling gives a score of 1.0 whereas non-positive scores are obtained for dissimilar clusterings (e.g., independent labeling). A previous study (Vinh et al. 2010) that conducted a useful survey can be referred for mathematical details.

- *Silhouette coefficient* For the case when ground truth class labeling is not available, clustering quality is measured using clusters themselves. Silhouette coefficient (Rousseeuw 1987) is such a measure. For a sample, it is given by the equation:

$$s = \frac{B - A}{max(A, B)} \tag{34}$$

where

- A is the average distance between a sample and all other points of the same cluster.
- B is the average distance between a sample and all other points of the next nearest cluster.

For dense and well-separated clusters, the score reaches a value of 1.0 and for incorrect clusters, it reached a value of − 1.0. The value of 0 indicates overlapping clusters.

5.4 Result analysis

To determine how the performance of different algorithms varies in comparison with each other when one goes from low dimensional to very high dimensional datasets, bar charts of AMI values obtained from different clusterings are plotted and shown in Figs. 6, 7 and

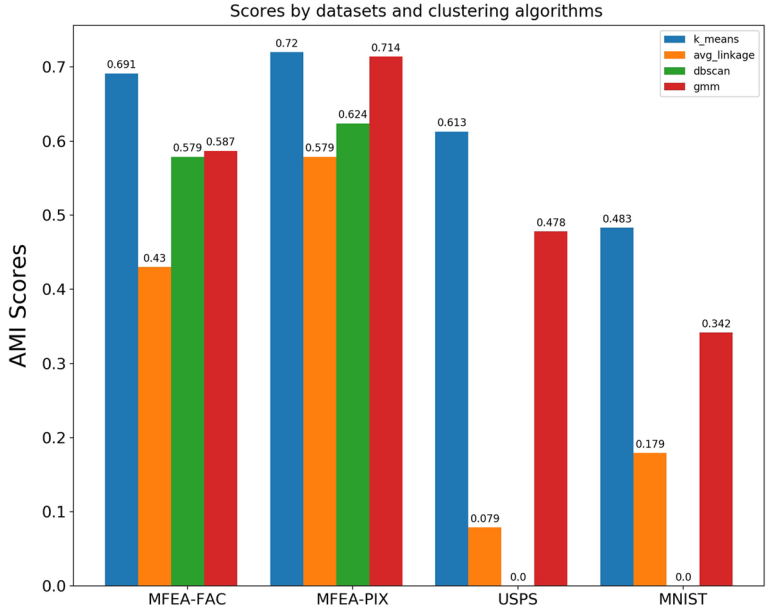


Fig. 7 Clustering results on medium dimensional datasets

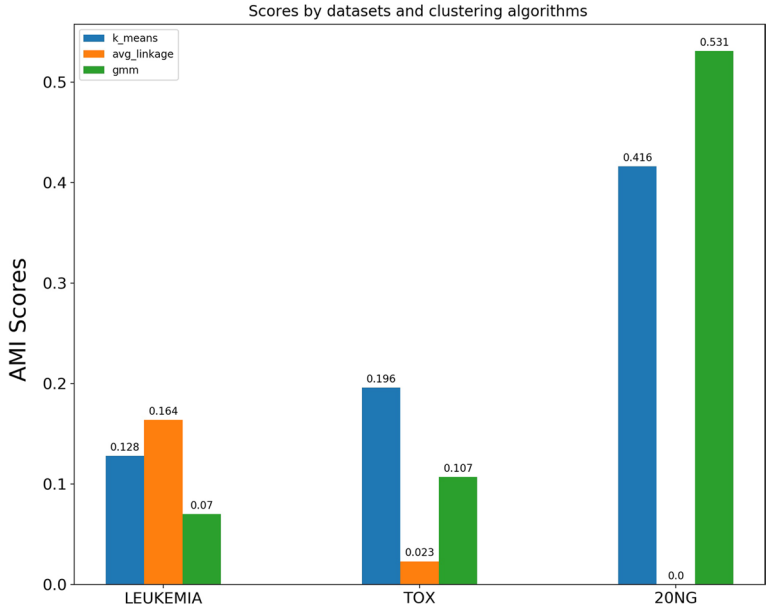


Fig. 8 Clustering results on high dimensional datasets

8. When a dataset goes from low to very high dimensions, the overall performance of all algorithms decreases. For example, the maximum AMI value in low dimensional datasets is 0.897 and that in high dimensional datasets is 0.531, both of which belong to GMM

clustering. This type of a decreasing trend is exhibited by every other clustering algorithm as well. DBSCAN and average linkage algorithms both fail to cluster high-dimensional datasets. Individual analysis of these three bar chart figures is given as follows:

Low dimensionality clustering On an average, K-means show the best performance with an average AMI value of 0.624 which is almost equal to the average AMI achieved by GMM (i.e., 0.622). Average linkage hierarchical clustering and density-based DBSCAN achieved average AMI values of 0.445 and 0.480 respectively.

Medium dimensionality clustering Regarding clustering of medium dimensionality datasets, on an average, the best AMI value is again achieved by k-means as 0.626, followed again by GMM with an average AMI value of 0.530. Average linkage hierarchical and DBSCAN algorithms achieved much lower values of 0.316 and 0.300, respectively.

High dimensionality clustering Finally, when the number of dimensions goes in thousands, all algorithms perform poorly as shown by very low average AMI values of 0.246 for K-means, 0.062 for average linkage, and 0.236 for GMM. DBSCAN fails to produce any clusters.

To determine the effect of using different proximity measures on partitioning and hierarchical clustering algorithms, Tables 4 and 5 list AMI values obtained using different proximity measures with partitioning algorithm k-means and average linkage hierarchical clustering algorithm, respectively. The best values are shown in bold. In low dimension clustering (≤ 100) with k-means, the best average AMI value of 0.626 is achieved using Bray–Curtis proximity measure. It is followed by Euclidean and Minkowski with an AMI value of 0.624. Going to the next level of medium dimensionality (101–1000) clustering with k-means, using the correlation proximity measure gives the best average AMI value of 0.645. Clustering of high dimensionality data (i.e. >1000 dimensions) using k-means, cosine distance achieves the best average AMI value of 0.332 followed by the correlation measure with a value of 0.322.

By using the average linkage clustering, the best average AMI value of 0.485 is achieved using the city-block proximity measure in low dimensional datasets. Going further to medium and high dimensionality datasets, the best average AMI value in both is achieved using the correlation-based distance of 0.365 and 0.255, respectively. Based on results obtained, it can be concluded that correlation distance shows the best performance in both the algorithms when the dataset consists of a large number of dimensions.

6 Conclusion

In this study, various proximity measures are reviewed and analyzed with respect to important aspects of data analysis such as sparsity, the existence of correlation among features and the effect of different feature scales. In addition, a theoretical procedure is proposed that can help researchers to select a versatile proximity measure for the clustering purpose. Then, a variety of clustering algorithm categories are reviewed and an experimental comparison has also been performed between the algorithms to analyze their performance. This is especially conducted concerning the dimensionality of datasets and it has been concluded that the performance of all traditional clustering algorithms is adversely affected when the number of dimensions is considerably high. Algorithm such as DBSCAN just fail to perform clustering when the dataset consisted of a very high number of dimensions. An experimental comparison to analyze the effect of using different proximity measures in a partitional clustering algorithm namely k-means and a hierarchical clustering algorithm

Table 4 AMI values of K-means clustering achieved on different datasets (classified on the basis of the number of dimensions)

Proximity measure	Low dimensional datasets				Average AMI	Medium dimensional datasets				Average AMI	High dimensional datasets			Average AMI
	IRIS (4)	CANCER (9)	GLASS (9)	MFEA-FOU (76)		MFEA-FAC (216)	MFEA-PIX (240)	USPS (256)	MNIST (784)		TOX (5748)	LEUKEMIA (7070)	20NEWSGROUP (50,000)	
Euclidean	0.748	0.725	0.346	0.678	0.624	0.691	0.720	0.538	0.483	0.608	0.196	0.128	0.416	0.246
Cosine	0.912	0.242	0.368	0.619	0.535	0.698	0.741	0.594	0.520	0.638	0.242	0.128	0.626	**0.332**
Correlation	0.861	0.287	0.368	0.623	0.534	0.682	0.741	0.647	0.513	**0.645**	0.238	0.110	0.618	0.322
Minkowski	0.748	0.725	0.346	0.678	0.624	0.691	0.720	0.613	0.483	0.626	0.196	0.128	0.416	0.246
Cityblock	0.740	0.661	0.338	0.630	0.592	0.696	0.707	0.500	0.417	0.580	0.234	0.155	0.043	0.144
Chebyshev	0.728	0.749	0.361	0.645	0.620	0.592	0.009	0.472	0.279	0.338	0.112	0.004	0.170	0.095
Canberra	0.845	0.712	0.248	0.498	0.575	0.661	0.703	0.419	N.A	0.594	0.238	0.161	0	0.133
Bray–curtis	0.725	0.797	0.338	0.647	**0.626**	0.688	0.739	0.477	0.507	0.602	0.226	0.136	0.527	0.296

Table 5 AMI values of average-linkage clustering achieved on different datasets (classified on the basis of the number of dimensions)

Proximity measure	Low dimensional datasets				Average AMI	Medium dimensional datasets				Average AMI	High dimensional datasets			Average AMI
	IRIS (4)	CANCER (9)	GLASS (9)	MFEA-FOU (76)		MFEA-FAC (216)	MFEA-PIX (240)	USPS (256)	MNIST (784)		TOX (5748)	LEUKEMIA (7070)	20NEWSGROUP (50,000)	
Euclidean	0.793	0.648	0.035	0.306	0.445	0.430	0.579	0.079	0.179	0.316	0.023	0.164	0.0	0.062
Cosine	0.574	0.001	0.229	0.455	0.314	0.662	0.778	0.082	0.015	0.284	0.334	0.154	0.001	0.163
Correlation	0.834	N.A	0.292	0.502	0.410	0.693	0.579	0.167	0.021	**0.365**	0.343	0.164	0.260	**0.255**
Minkowski	0.793	0.648	0.035	0.306	0.445	0.430	0.581	0.079	0.179	0.317	0.023	0.164	0.0	0.062
Cityblock	0.767	0.742	0.101	0.332	**0.485**	0.485	0.512	0.066	0.005	0.267	0.049	0.169	0.001	0.073
Chebyshev	0.620	0.001	0.227	0.527	0.343	0.144	0.001	0.005	0.001	0.037	0.003	0.120	0.0	0.041
Canberra	0.574	0.820	0.252	0.206	0.463	0.641	0.493	0.067	0.154	0.338	0.330	0.156	0.001	0.162
Bray–curtis	0.591	0.778	0.101	0.285	0.438	0.639	0.363	0.133	0.003	0.284	0.276	0.154	0.001	0.143

namely the average linkage, is conducted. The results showed that the average performance scores of clustering vary when a different proximity measure is used. In addition, the best performing measures have been reported.

References

Agrawal R, Gehrke J, Gunopulos D, Raghavan P (2005) Automatic subspace clustering of high dimensional data. Data Min Knowl Discov 11(1):5–33

Altınçay H, Erenel Z (2010) Analytical evaluation of term weighting schemes for text categorization. Pattern Recognit Lett 31(11):1310–1323

Ankerst M, Breunig MM, Kriegel HP, Sander J (1999) Optics: ordering points to identify the clustering structure. In: ACM Sigmod record, vol 28. ACM, pp 49–60

Basu T, Murthy C (2015) A similarity assessment technique for effective grouping of documents. Inf Sci 311:149–162

Bezdek JC (1981) Objective function clustering. In: Pattern recognition with fuzzy objective function algorithms. Springer, pp 43–93

Bezdek JC, Ehrlich R, Full W (1984) FCM: the fuzzy c-means clustering algorithm. Comput Geosci 10(2–3):191–203

Bouchachia A, Pedrycz W (2006) Enhancement of fuzzy clustering by mechanisms of partial supervision. Fuzzy Sets Syst 157(13):1733–1759

Cambria E, Mazzocco T, Hussain A, Eckl C (2011) Sentic medoids: organizing affective common sense knowledge in a multi-dimensional vector space. In: International symposium on neural networks. Springer, pp 601–610

Cambria E, Fu J, Bisio F, Poria S (2015) Affective space 2: enabling affective intuition for concept-level sentiment analysis. In: Twenty-ninth AAAI conference on artificial intelligence, pp 508–514

Cetinkaya S, Basaraner M, Burghardt D (2015) Proximity-based grouping of buildings in urban blocks: a comparison of four algorithms. Geocarto Int 30(6):618–632

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B (Methodol) 39:1–38

Dunn JC (1973) A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. J Cybern 3:32–57

Ester M, Kriegel HP, Sander J, Xu X et al (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. Kdd 96:226–231

Fahad A, Alshatri N, Tari Z, Alamri A, Khalil I, Zomaya AY, Foufou S, Bouras A (2014) A survey of clustering algorithms for big data: taxonomy and empirical analysis. IEEE Trans Emerg Top Comput 2(3):267–279

Fisher DH (1987) Knowledge acquisition via incremental conceptual clustering. Mach Learn 2(2):139–172

García-Pablos A, Cuadros M, Rigau G (2018) W2VLDA: almost unsupervised system for aspect based sentiment analysis. Expert Syst Appl 91:127–137

Gennari JH, Langley P, Fisher D (1989) Models of incremental concept formation. Artif Intell 40(1–3):11–61

Glen S. Bray curtis dissimilarity. http://www.statisticshowto.com/bray-curtis-dissimilarity/. Accessed 28 Apr 2018

Glen S. Kullback–leibler kl divergence. https://www.statisticshowto.datasciencecentral.com/kl-divergence. Accessed 28 Apr 2018

Guha S, Rastogi R, Shim K (1998) Cure: an efficient clustering algorithm for large databases. In: ACM sigmod record, vol 27. ACM, pp 73–84

Guha S, Rastogi R, Shim K (2000) Rock: a robust clustering algorithm for categorical attributes. Inf Syst 25(5):345–366

Gustafson DE, Kessel WC (1979) Fuzzy clustering with a fuzzy covariance matrix. In: 1978 IEEE conference on decision and control including the 17th symposium on adaptive processes. IEEE, pp 761–766

Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques. Elsevier, Amsterdam

Han X, Quan L, Xiong X, Almeter M, Xiang J, Lan Y (2017) A novel data clustering algorithm based on modified gravitational search algorithm. Eng Appl Artif Intell 61:1–7

Hanna AR, Rao C, Athanasiou T (2010) Graphs in statistical analysis. In: Key topics in surgical research and methodology. Springer, pp 441–475

Hinneburg A, Keim DA (1999) Optimal grid-clustering: towards breaking the curse of dimensionality in high-dimensional clustering. In: Proceedings of the 25th international conference on very large databases, 1999, pp 506–517

Hinneburg A, Keim DA et al (1998) An efficient approach to clustering in large multimedia databases with noise. KDD 98:58–65
Hong X, Yu Z, Tang M, Xian Y (2017) Cross-lingual event-centered news clustering based on elements semantic correlations of different news. Multimed Tools Appl 76(23):25129–25143
Huang Z (1997) A fast clustering algorithm to cluster very large categorical data sets in data mining. DMKD 3(8):34–39
Huang A (2008) Similarity measures for text document clustering. In: Proceedings of the sixth New Zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand, pp 49–56
Jaccard index (2018). https://en.wikipedia.org/wiki/Jaccard_index. Accessed 28 Apr 2018
Jain AK (2010) Data clustering: 50 years beyond k-means. Pattern Recognit Lett 31(8):651–666
Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice Hall, Englewood Cliffs
Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv (CSUR) 31(3):264–323
Jan TG (2020) Clustering of tweets: a novel approach to label the unlabelled tweets. In: Proceedings of ICRIC 2019. Springer, pp 671–685
Kameshwaran K, Malarvizhi K (2014) Survey on clustering techniques in data mining. Int J Comput Sci Inf Technol 5(2):2272–2276
Kannan S, Ramathilagam S, Devi R, Hines E (2012) Strong fuzzy c-means in medical image data analysis. J Syst Softw 85(11):2425–2438
Karypis G, Han EH, Kumar V (1999) Chameleon: hierarchical clustering using dynamic modeling. Computer 32(8):68–75
Kohonen T (1998) The self-organizing map. Neurocomputing 21(1–3):1–6
Kruse R, Döring C, Lesot MJ (2007) Fundamentals of fuzzy clustering. In: de Oliveira JV, Pedrycz W (eds) Advances in Fuzzy Clustering and its Applications. Wiley, Chichester, pp 3–30
Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22(1):79–86
Lai DTC, Garibaldi JM (2011) A comparison of distance-based semi-supervised fuzzy c-means clustering algorithms. In: 2011 IEEE international conference on fuzzy systems (FUZZ). IEEE, pp 1580–1586
Lan M, Sung SY, Low HB, Tan CL (2005) A comparative study on term weighting schemes for text categorization. In: Proceedings. 2005 IEEE international joint conference on neural networks, 2005., vol 1. IEEE, pp 546–551
Leoncini A, Sangiacomo F, Peretti C, Argentesi S, Zunino R, Cambria E (2011) Semantic models for style-based text clustering. In: 2011 IEEE fifth international conference on semantic computing. IEEE, pp 75–82
Li C, Liu L, Jiang W (2008) Objective function of semi-supervised fuzzy c-means clustering algorithm. In: 6th IEEE international conference on industrial informatics, 2008. INDIN 2008. IEEE, pp 737–742
Lin YS, Jiang JY, Lee SJ (2014) A similarity measure for text classification and clustering. IEEE Trans Knowl Data Eng 26(7):1575–1590
MacQueen J et al (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol 1. Oakland, CA, USA, pp 281–297
Manning CD, Manning CD, Schütze H (1999) Foundations of statistical natural language processing. MIT Press, Cambridge
McCune B, Grace JB, Urban DL (2002) Analysis of ecological communities, vol 28. MjM Software Design, Gleneden Beach
Montoyo A, MartíNez-Barco P, Balahur A (2012) Subjectivity and sentiment analysis: an overview of the current state of the area and envisaged developments. Decis Support Syst 53:675–689
Nanda SJ, Panda G (2014) A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm Evol Comput 16:1–18
Ng RT, Han J (1994) Efficient and effective clustering methods for spatial data mining. In: Proceedings of VLDB, pp 144–155
Ng RT, Han J (2002) Clarans: a method for clustering objects for spatial data mining. IEEE Trans Knowl Data Eng 14(5):1003–1016
Park HS, Jun CH (2009) A simple and fast algorithm for k-medoids clustering. Expert Syst Appl 36(2):3336–3341
Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
Pedrycz W, Waletzky J (1997) Fuzzy clustering with partial supervision. IEEE Trans Syst Man Cybern Part B (Cybern) 27(5):787–795
Ravi K, Ravi V (2015) A survey on opinion mining and sentiment analysis: tasks, approaches and applications. Knowl Based Syst 89:14–46

Ross TJ (2005) Fuzzy logic with engineering applications. Wiley, Hoboken
Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
Rudolf Kruse Christian Döring ML (2007) Fundamentals of fuzzy clustering. In: de Oliveira WP J Valente (ed) Advances in fuzzy clustering and its applications. Wiley, Oxford, pp 3–30 chap. 1
Saraçoğlu R, Tütüncü K, Allahverdi N (2007) A fuzzy clustering approach for finding similar documents using a novel similarity measure. Expert Syst Appl 33(3):600–605
Schoenharl TW, Madey G (2008) Evaluation of measurement techniques for the validation of agent-based simulations against streaming data. In: International conference on computational science. Springer, pp 6–15
Schubert E, Sander J, Ester M, Kriegel HP, Xu X (2017) DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. ACM Trans Database Syst (TODS) 42(3):1–21
Sedding J, Kazakov D (2004) Wordnet-based text document clustering. In: proceedings of the 3rd workshop on robust methods in analysis of natural language data. Association for Computational Linguistics, pp 104–113
Sehgal G, Garg DK (2014) Comparison of various clustering algorithms. Int J Comput Sci Inf Technol 5(3):3074–3076
Selim SZ, Alsultan K (1991) A simulated annealing algorithm for the clustering problem. Pattern Recognit 24(10):1003–1008
Sheikholeslami G, Chatterjee S, Zhang A (1998) Wavecluster: a multi-resolution clustering approach for very large spatial databases. VLDB 98:428–439
Shirkhorshidi AS, Aghabozorgi S, Wah TY, Herawan T (2014) Big data clustering: a review. In: International conference on computational science and its applications. Springer, pp 707–720
Shirkhorshidi AS, Aghabozorgi S, Wah TY (2015) A comparison study on similarity and dissimilarity measures in clustering continuous data. PLoS ONE 10(12):e0144059
Strehl A, Ghosh J, Mooney R (2000) Impact of similarity measures on web-page clustering. In: Workshop on artificial intelligence for web search (AAAI 2000), vol 58, pp 58–64
Tang G, Xia Y, Cambria E, Jin P, Zheng TF (2015) Document representation with statistical word senses in cross-lingual document clustering. Int J Pattern Recognit Artif Intell 29(02):1559003
Vinh NX, Epps J, Bailey J (2010) Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J Mach Learn Res 11:2837–2854
Vossen P (2002) Eurowordnet general document version 3. University of Amsterdam, Amsterdam
Wang W, Yang J, Muntz R et al (1997) Sting: a statistical information grid approach to spatial data mining. VLDB 97:186–195
Wei T, Lu Y, Chang H, Zhou Q, Bao X (2015) A semantic approach for text clustering using wordnet and lexical chains. Expert Syst Appl 42(4):2264–2275
Wu Zd, Xie Wx, Yu Jp (2003) Fuzzy c-means clustering algorithm based on kernel method. In: Proceedings fifth international conference on computational intelligence and multimedia applications. ICCIMA 2003. IEEE, pp 49–54
Xia Y, Tang N, Hussain A, Cambria E (2015) Discriminative bi-term topic model for headline-based social news clustering. In: The twenty-eighth international flairs conference, pp 311–316
Xie J, Girshick R, Farhadi A (2016) Unsupervised deep embedding for clustering analysis. In: International conference on machine learning, pp 478–487
Xu D, Tian Y (2015) A comprehensive survey of clustering algorithms. Ann Data Sci 2(2):165–193
Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16(3):645–678
Xu X, Ester M, Kriegel HP, Sander J (1998) A distribution-based clustering algorithm for mining in large spatial databases. In: 14th international conference on data engineering, 1998. Proceedings. IEEE, pp 324–331
Yan X, Guo J, Lan Y, Cheng X (2013) A biterm topic model for short texts. In: Proceedings of the 22nd international conference on World Wide Web. ACM, pp 1445–1456
Yasunori E, Yukihiro H, Makito Y, Sadaaki M (2009) On semi-supervised fuzzy c-means clustering. In: 2009 IEEE international conference on fuzzy systems. IEEE, pp 1119–1124
Zhang T, Ramakrishnan R, Livny M (1996) Birch: an efficient data clustering method for very large databases. In: ACM sigmod record, vol 25. ACM, pp 103–114
Zhang D, Tan K, Chen S (2004) Semi-supervised kernel-based fuzzy c-means. In: International conference on neural information processing. Springer, pp 1229–1234