Methods for Clustering Categorical and Mixed Data: An Overview and New Algorithms

Sadaaki Miyamoto[1(✉)], Van-Nam Huynh[2], and Shuhei Fujiwara[1]

[1] University of Tsukuba, Tsukuba, Japan
miyamoto.sadaaki.fu@u.tsukuba.ac.jp
[2] Japan Advanced Institute of Science and Technology, Nomi, Japan
huynh@jaist.ac.jp

Abstract. Methods of clustering for categorical and mixed data are considered. Dissimilarities for this purpose are reviewed and different classes of algorithms according to different classes of similarities are discussed. Details of several algorithms are then given, which include agglomerative hierarchical clustering, K-means and related methods such as K-medoids and K-modes, and methods of network clustering. The way how the combinations of existing ideas leads to new algorithms is discussed.

Keywords: Data clustering · Categorical data · Mixed data
Network clustering · K-medoids

1 Introduction

Data clustering has now become a standard technique of data mining, and yet it has a number of unique characteristics different from other methods of supervised and unsupervised classification. One of those characteristics is that different types of data are assumed to be given for analysis: not only the Euclidean space but also other spaces and moreover general types of dissimilarities can be used as measures of relatedness between a pair of objects. On the other hand, a standard class of clustering algorithms of *agglomerative hierarchical clustering* has an unique and useful form of output that is called a *dendrogram*. The dendrogram is popular in various fields of applications and its usefulness could not be ignored.

In this paper we give a brief overview of methods of clustering for non-Euclidean models in the sense that given data types of an object for clustering is categorical or mixed; a mixed data type consists of categorical data and numerical data at the same time. First dissimilarities for categorical and mixed data are discussed. Then three classes of methods of clustering are introduced, which are agglomerative hierarchical clustering, non-hierarchical methods for Euclidean data, and non-hierarchical methods for non-Euclidean data. The best known method of K-means clustering is in the second class, related methods of the third class is considered which includes K-median, K-modes, and K-medoids.

V.-N. Huynh et al. (Eds.): IUKM 2018, LNAI 10758, pp. 75–86, 2018.
https://doi.org/10.1007/978-3-319-75429-1_7

Consideration of these methods stimulates the development of new linkage methods in the first class. These considerations then lead us to related algorithms such as a generative model and a method of fuzzy clustering. Moreover network clustering is briefly mentioned. The way how these methods lead to the developemnt of new methods for categorical and mixed data is discussed.

2 Categorical Data and Dissimilarities

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ be a finite set of objects for clustering. Assume that $\mathcal{A} = \{A_1, \ldots, A_M\}$ be another set of attributes. For each A_j, an associated set Z_j which contains all values that A_j takes. Thus, every $z \in Z_j$ is a value of attribute A_j. For each x_i and A_j, the value of object x_i concerning attribute A_j is given by $x_i^j \in Z_j$. We write $A_j(x_i) = x_i^j$, as A_j is a mapping ($A_j \colon \mathcal{X} \to Z_j$). Alternatively, we express $x = (x^1, \ldots, x^M)$ as a kind of vectors, although its components are not necessarily numbers. Concretely, Z_j can be *numerical* when its elements are numerical values: $Z_j \subseteq \boldsymbol{R}$. Or Z_j can be *symbolical* when its elements are symbols and not numerical. Note also that a symbol represents a category, and hence the words 'categorical' and 'symbolical' are used for the same meaning herein. Let us write a typical case as $Z_j = \{t_1, \ldots, t_q\}$ where the elements t_l are symbols.

Assume that clusters denoted by $G_1, \ldots, G_K$ are disjoint subsets of $\mathcal{X}$ such that the union of clusters covers the whole set:

$$\bigcup_{i=1}^{K} G_i = \mathcal{X}, \quad G_i \cap G_j = \emptyset \ \ (i \neq j). \tag{1}$$

Moreover the collection of clusters is denoted by $\mathcal{G} = \{G_1, \ldots, G_K\}$.

Similarity or dissimilarity is a key concept in data clustering. We assume a similarity measure $s(x, x')$ or a dissimilarity measure $d(x, x')$ is defined between a pair of objects $x, x' \in \mathcal{X}$. The difference between similarity and dissimilarity is that two objects are similar or near when similarity between them has a high value while they are similar when dissimilarity value is lower. Accordingly,

$$s(x, x) = \max_{x' \in \mathcal{X}} s(x, x'), \quad d(x, x) = \min_{x' \in \mathcal{X}} d(x, x').$$

Symmetric property is also assumed for the both measures:

$$s(x, x') = s(x', x), \quad d(x, x') = d(x', x). \tag{2}$$

Note that the triangular inequality is not assumed: the triangular inequality in general is not especially useful in clustering.

2.1 Measures of Dissimilarity

We mostly use dissimilarity and refer to similarity only when necessary. How to define an appropriate dissimilarity is a first problem to be considered in clustering. Sometimes $d(x, x')$ is directly given without referring to their attributes,

as in the case of network clustering [5,20,21]. We, however, assume that dissimilarities are defined by the observation of attribute values x_i^j. Since we have different types of sets Z_j in general, different kinds of dissimilarities should be considered. We therefore assume that $d_j(x, y)$ for $x, y \in Z_j$, and consider how d_j should be defined.

Let us consider the most frequent case of an Euclidean space $\boldsymbol{R}^M$. In this case $x, y \in \boldsymbol{R}$ for all attributes and we set

$$d_j(x, y) = (x - y)^2, \quad \textit{for all } 1 \leq j \leq M,$$

and the dissimilarity is given by

$$d(x, x') = \sum_{j=1}^{M} d_j(x^j, y^j) = \sum_{j=1}^{M} (x^j - y^j)^2, \tag{3}$$

for $x = (x^1, \ldots, x^M)$ and $x' = (y^1, \ldots, y^M)$. We also write $d(x, x') = \|x - x'\|^2$ using the Euclidean norm symbol. Note that the squared Euclidean norm is used instead of the norm itself.

Let us suppose that data are of a mixed type, i.e., some Z_j is numerical while another Z_l is symbolical. A simple definition of a dissimilarity is that

$$d_j(x, y) = \frac{1}{2}|x - y|, \tag{4}$$

i.e., $d_j(x, y)$ is the difference between the two numerical values, while

$$d_l(t_h, t_k) = \begin{cases} 1 & (t_h \neq t_k), \\ 0 & (t_h = t_k). \end{cases} \tag{5}$$

and define

$$d(x, x') = \sum_{j=1}^{M} d_j(x^j, y^j). \tag{6}$$

Let us assume that there is no Z_l of the set of symbols, then all attribute values are numerical but we do not have a squared Euclidean dissimilarity, but instead we have the L_1-norm:

$$d(x, x') = \frac{1}{2} \sum_{j=1}^{M} |x^j - y^j| = \frac{1}{2} \|x - y\|_{L_1}.$$

On the other hand, if all attribute values are symbolic, Eq. (6) consists of (5) alone. There is an interesting relationship between the latter two. Let us convert x into 0/1 numerical values. Actually, only one of $t_1, \ldots, t_M$, say t_1, represents the object and hence we can write $x = \{t_1\}$ or $x = (1, 0, \ldots, 0)$ using 0 and 1. Suppose $x' = \{t_2\}$ or $x' = (0, 1, 0, \ldots, 0)$. Then it is easy to see that

$$d_l(t_1, t_2) = \frac{1}{2}|x - x'|$$

If all attributes are symbolical, we have $d(x, x') = \frac{1}{2}\|x - x'\|_{L_1}$. Thus, the weight $\frac{1}{2}$ in (4) is justified.

2.2 Minimization Problems

For later use, we consider optimization problems: $\min_{x \in \boldsymbol{R}^M} \sum_{y \in \mathcal{X}} d(x, y)$, in case when all values are numerical, and we assume the both cases of an Euclidean space (3) and L_1-space (6).

In the Euclidean space, it is easy to see that the solution is the average: $x = \dfrac{1}{|\mathcal{X}|} \sum_{y \in \mathcal{X}} y$. where $|\mathcal{X}|$ is the number of elements in $\mathcal{X}$.

On the other hand, if L_1-space is used, the solution is the median. Each component of the median is defined independently. Let the first component (corresponding to A_1) is $x_1^1, x_2^1, \ldots, x_M^1$. Sort this set of real numbers into ascending order and the result is $y_1, \leq y_2 \leq \cdots \leq y_M$. Then the median for the first component is $y_{[M/2]+1}$. Other components are calculated in the same manner.

There is still other minimization problems. Suppose all data are symbolic, we consider

$$\min_{x \in \mathcal{Z}} \sum_{x' \in \mathcal{X}} d(x, x'), \tag{7}$$

where $\mathcal{Z} = Z_1 \times \cdots \times Z_M$. Note that $d(x, x')$ is defined by (6) and (5). To solve this problem, let the frequency of occurrences of $y_k \in Z_j$ be f_k on $\mathcal{Z}$. Thus we have a histogram $(f_1/y_1, \ldots, f_L/y_L)$ for X_j. Assume that the maximum of $f_1, \ldots, f_L$ is f_h, then the mode is written as

$$\text{mode}(\mathcal{X}, Z_j) = (\arg\max\{f_1, \ldots, f_L\}, \max\{f_1, \ldots, f_L\}) = (h, f_h), \tag{8}$$

$$\arg \text{mode}(\mathcal{X}, Z_j) = h, \tag{9}$$

$$\text{value mode}(\mathcal{X}, Z_j) = f_h. \tag{10}$$

Then it is easy to see that the solution of (7) is given by

$$(\text{mode}(\mathcal{X}, Z_1), \ldots, \text{mode}(\mathcal{X}, Z_M)).$$

These minimization problems with their solutions are useful in considering K-modes and related clustering problems.

3 Algorithms of Clustering

Two major methods are agglomerative hierarchical clustering and the K-means.

3.1 Agglomerative Hierarchical Clustering

The agglomerative hierarchical algorithm [1,10,18] is one of best known methods of clustering. It uses a measure $d(G_i, G_j)$ of an inter-cluster dissimilarity. The following is a general description of the agglomerative hierarchical algorithm [18]. Note that initial clusters $\mathcal{G}(0) = \{G(0)_1, \ldots, G(0)_{C_0}\}$ are assumed to be given. Typically, $G(0)_i = \{x_i\}$, but we assume other cases later.

AHC (Agglomerative Hierarchical Clustering)
AHC1: Each object forms an initial cluster: $G_i = G(0)_i,\ (i = 1, \cdots, N)$. $C = N$, (C is the number of clusters). For all $G_i, G_j \in \mathcal{G}$, let $d(G_i, G_j) = d(x_i, x_j)$.
AHC2: Find the pair of clusters of minimum dissimilarity:

$$(G_q, G_r) = \arg \min_{G_i, G_j \in \mathcal{G}} d(G_i, G_j) \tag{11}$$

$$m_C = d(G_q, G_r) \tag{12}$$

Add $G' = G_q \cup G_r$ to $\mathcal{G}$ and delete G_q, G_r from $\mathcal{G}$. Let $C = C - 1$. If $C = 1$, output clusters as a *dendrogram* and stop.
AHC3: Update dissimilarity $d(G, G')$ between the merged cluster G' and all other clusters $G \in \mathcal{G}$. Go to **AHC2**.
End of AHC.

Here, $m_N, \ldots, m_2$ are called the levels of merging clusters.

We have several *linkage methods* to update dissimilarity $d(G, G')$ in **AHC3**, from which the single linkage, the average linkage, and the Ward method are mentioned here.

Single linkage: $d(G_i, G_j) = \min_{x \in G_i, y \in G_j} d(x, y)$.

Average linkage: $d(G_i, G_j) = \dfrac{1}{|G_i||G_j|} \displaystyle\sum_{x \in G_i, y \in G_j} d(x, y)$.

Ward method: Assume

$$E(G) = \sum_{x_k \in G} \|x_k - M(G)\|^2.$$

Let

$$d(G_i, G_j) = E(G_i \cup G_j) - E(G_i) - E(G_j).$$

where $M(G)$ is the centroid of G: $M(G) = \displaystyle\sum_{x_k \in G} \frac{x_k}{|G|}$, and $\|\cdot\|$ is the Euclidean norm: this method assumes that the objects are points in an Euclidean space.

They moreover use the following formulas of updating in **AHC3** in which $d(G, G')$ is expressed using $d(G, G_q)$, $d(G, G_r)$, and so on.

Updating formula of the single linkage:

$$d(G, G') = \min\{d(G, G_q), d(G, G_r)\}.$$

Updating formula of the Ward method:

$$d(G, G') = \frac{(|G_q| + |G|)d(G_q, G) + (|G_r| + |G|)d(G_r, G) - |G|d(G_q, G_r)}{|G_q| + |G_r| + |G|}.$$

The updating formula of the average linkage is omitted here. See, e.g., [18] for more detail.

The single linkage and the Ward method are two popular algorithms in agglomerative hierarchical clustering. The former is known to have best theoretical properties [18], while the Ward method has been considered to be practically useful by researchers in applications.

3.2 The K-means and Related Methods

We assume that objects $x_1, \ldots, x_N$ are in a space $\mathbf{S}$ whose distance is defined by the dissimilarity $d(x, y)$. Consider the next problem of *alternate minimization*.

K-means prototype algorithm.
Step 0. Give an initial partition $G_1, \ldots, G_K$ of $\{x_1, \ldots, x_N\} \subseteq \mathbf{S}$.
Step 1. Let

$$v_i = \arg\min_{v \in \mathbf{S}} \sum_{x_k \in G_i} d(x_k, v), \quad i = 1, 2, \ldots, K. \tag{13}$$

Step 2. Allocate each x_k $(k = 1, \ldots, N)$ to the cluster of the nearest center v_i:

$$x_k \to G_i \iff v_i = \arg\min_{1 \le j \le K} d(x_k, v_j). \tag{14}$$

Step 3. If $(v_1, \ldots, v_K)$ is convergent, stop. Else go to **step 1**.
End K-means prototype.

The above algorithm describes a family of different methods.

The method of K-means [15] is the most popular clustering algorithm. It assumes that the objects $x_1, \ldots, x_N$ are points in an Euclidean space. Hence we assume $\mathbf{S} = \boldsymbol{R}^p$ with $d(x, y) = \|x - y\|^2$. Accordingly the center of a cluster (13) is the centroid:

$$v_i = M(G_i) = \frac{1}{|G_i|} \sum_{x_k \in G_i} x_k.$$

Thus the K-means prototype algorithm is reduced to the K-means algorithm.

K-median and K-mode algorithms are derived likewise. If L_1-space is used, then v_i is given by the median described above; if the data are categorical and the dissimilarity is given by (6), then the center v_i is given by the mode for cluster G_i:

$$v_i^j = \begin{cases} 1 & (j = \arg\,\mathrm{mode}(G_i, Z_j)), \\ 0 & (\text{otherwise}). \end{cases} \tag{15}$$

and we have the K-mode algorithm.

Moreover if we have mixed data in which numerical data has L_1-norm, then the resulting algorithm has the mixture of the median and the mode corresponding to the data types.

There is still another method of the K-means family, in which $\mathbf{S}$ is the set of objects itself: $\mathbf{S} = \mathcal{X}$ with the general dissimilarity $d(x, x')$. In such a case the space is a weighted network and accordingly the element v_i corresponds to an object which satisfies

$$v_i = \arg\min_{v \in \mathcal{X}} \sum_{x_k \in G_i} d(x_k, v). \tag{16}$$

The above defined object for G_i is called the medoid [14] for cluster G_i. Thus the algorithm gives the method of K-medoids. It is obvious to see $v_i \in G_i$.

3.3 Network Clustering

The last method of K-medoids is an algorithm of network clustering in the sense that no other space than just the weighted network is given. There are other methods that should be mentioned in addition.

DBSCAN [9] is known to be an efficient algorithm that searches clusters of *core-points* on a weighted graph. This method has been proved to be a variation of the single linkage that connects core-points and node-points [16].

Newman's method [20,21] of hierarchical clustering and its non-hierarchical version [5] use the modularity index in a network; they automatically determine the number of clusters by optimizing the index. It seems that the modularity index works effectively in the both algorithms, but the non-hierarchical algorithm is faster and appropriate for handling large-scale data sets. On the other hand, the hierarchical version can output a dendrogram, but the shape of the dendrogram by this method is very different from those by the traditional linkage method, as we will see later in an example, and Newman's method may not be useful in understanding subcluster structures in a dendrogram.

3.4 Fuzzy Clustering

The method of fuzzy c-means [3,4,8,12,19] has been popular among researchers in at least two senses. First, the method gives fuzzy clusters instead of crisp clusters with much more information on the belongingness of an object to a cluster. Second, the algorithm is known to have high robustness over the K-means algorithm as to the variation of initial values and also noises and outliers. The robustness concerning outliers may still be improved by using fuzzy clustering and noise clustering [6,7].

Moreover the method of fuzzy c-means using an entropy term generalizes the Gaussian mixture model (see, e.g., [19]) and thus shows the expressive power of the fuzzy clustering model. Recently, Honda *et al.* [12] showed the multinomial mixture model for categorical data can be generalized by using a fuzzy co-clustering model.

4 Development of New Algorithms

We consider new algorithms on the basis of the above methods.

4.1 Fuzzy Clustering

Fuzzy clustering for categorical and mixed data can be studied by a similar way as the fuzzy c-means. The objective function is as follows:

$$J(U,V) = \sum_{i=1}^{c}\sum_{k=1}^{N}(u_{ki})^m d(x_k, v_i), \quad (m > 1). \tag{17}$$

where $d(x_k, v_i)$ is given by (6). U has the constraint: $\{u_{ki} \geq 0$ for all $k, i, \ \sum_{j=1}^{c} u_{kj} = 1$ for all $j\}$, while $V = (v_1, \ldots, v_c)$ does not have a constraint. The alternate minimization $\min_U J(U, V)$ and $\min_V J(U, V)$ while other variable is fixed to the last optimal solution is iteratively applied to $J(U, V)$ until convergence. There is no guarantee that the converged solution is the optimal solution for $J(U, V)$, but the solutions are empirically satisfactory.

For the present case of (17), the optimal solution U is:

$$u_{ki} = \frac{d(x_k, v_i)^{-\frac{1}{m-1}}}{\sum_{j=1}^{c} d(x_k, v_j)^{-\frac{1}{m-1}}}, \tag{18}$$

which is essentially the same as that for the standard Euclidean space, while the optimal solution V is different from the Euclidean case, and hence we should consider the case of L_1-space, that of categorical data, and that of medoids ($\mathbf{S} = \mathcal{X}$).

For $\mathbf{S} = \boldsymbol{R}^p$ with L_1 norm, we can use a weighted median algorithm [17]. For the case of medoids, the algorithm is essentially the same as the crisp case, i.e., we search the minimum of $\sum_k (u_{ki})^m d(x_k, v)$ with respect to v.

Since both a medoid and center are good representatives of a cluster, we can consider a new algorithm of the two representatives: Let $v_i = (v_i', v_i'')$ and assume that v_i' is a non-medoid center and v_i'' is a medoid, we define a new dissimilarity

$$d'(x_k, v_i) = \alpha d(x_k, v_i') + (1 - \alpha) d(x_k, v_i''), \tag{19}$$

with $0 < \alpha < 1$. If $d(x_k, v_i)$ is the L_1-distance, then v_i' is a weighted median and v_i'' is a medoid for G_i.

Such an algorithm using two representatives for a cluster have been developed for non-symmetric measure of dissimilarity [11,13]. Since we do not consider a non-symmetric measure here, we omit the detail.

4.2 Two-Stage Algorithms

A multi-stage algorithm can be a useful procedure when large-scale data should be handled. Consider a case when a large number of objects are gathered into a medium number of clusters using K-means, and then the centers are made into clusters using the same algorithm. In such a case K-medoids are also appropriate, since an object is made as a representative of a cluster.

Tamura *et al.* [22] proposed a two-stage procedure in which the first stage uses a p-pass K-means (i.e., a K-means procedure where the number of iterations is p; $p = 1$ or 2 is usually used.) in the first stage with the initial selection of centers using K-means++ [2], and the Ward method is used for the second stage. There is no loss of information because K-means and Ward method are based on the same criterion of the squared sum of errors from the center.

Two-stage algorithms of a median-Ward method and a medoid-Ward method can moreover be developed: the median-Ward method uses the one-pass K-median and an agglomerative procedure like the Ward method which uses L_1-norm throughout the procedure. The medoid-Ward method uses K-medoids in the first stage and an agglomerative procedure like the Ward method which uses an arbitrary dissimilarity.

In the latter method we use an objective function

$$J(\mathcal{G}) = \sum_{k=1}^{C} \sum_{x_l \in G_k} d(x_l, v_k),$$

where v_k is the medoid for G_k. Moreover we assume

$$\mathcal{G}[i,j] = \mathcal{G} \cup \{G_i \cup G_j\} - \{G_i\} - \{G_j\}, \quad d(G_i, G_j) = J(\mathcal{G}[i,j]) - J(\mathcal{G}).$$

The dissimilarity $d(G_i, G_j)$ is used in AHC algorithm of the medoid-Ward method. It is immediate to observe $d(G_i, G_j) \geq 0$. Note that if we set $d(x_l, v_k) = \|x_l - v_k\|_{L_1}$, we have the median-Ward method.

The last two of the median-Ward and the medoid-Ward algorithms have the drawback of much calculation in the second stage, but the number of objects are not many in the second stage and hence we can manage the processing time practically, but large-scale problems of millions of objects still have the fundamental problem of the inefficiency of calculations.

An Example: We show an example of network clustering on Twitter data [11]. Figure 1 shows the dendrogram output using Newman's method. The Twitter data of the graph with the number of nodes is 1744 and 108312 edges. The data consists of 5 political parties in Japan. The details are given in [11] and omitted here. The adjacency matrix A is made into the similarity matrix $S = A + A^2/2$. Apart from the large number of nodes, it is hard to observe subcluster structures in this dendrogram.

Figure 2 shows the dendrogram using the average linkage to the same data. The result shows subcluster structures but due to the large number of nodes, to observe the details is difficult.

Figure 3 shows the result using the two-stage procedure of medoid-Ward method. Subclusters are more clearly shown in this figure. The initial objects are summarized into 100 small clusters in the first stage.

Note that five clusters are observed in the latter two figures, while they are not clear in the first figure.

4.3 Use of Core Points

The concept of core points was introduced in DBSCAN [9], which is an important idea for effectively reducing the number of points for clustering.

In order to define a core point, a neighborhood $N(x;\epsilon)$ is defined: $N(x;\epsilon) = \{y \in \mathcal{X} \colon d(x,y) \leq \epsilon\}$. Let L be a positive integer. If $|N(x;\epsilon)| \geq L$ (the number

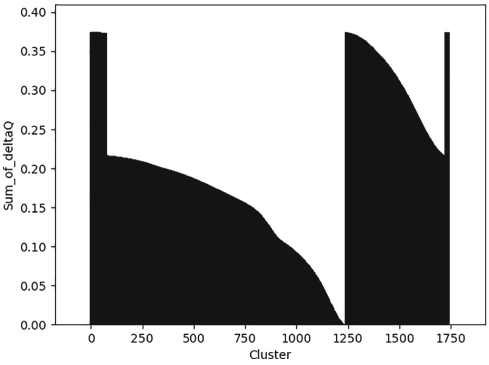

Fig. 1. Dendrogram from party data using newman method

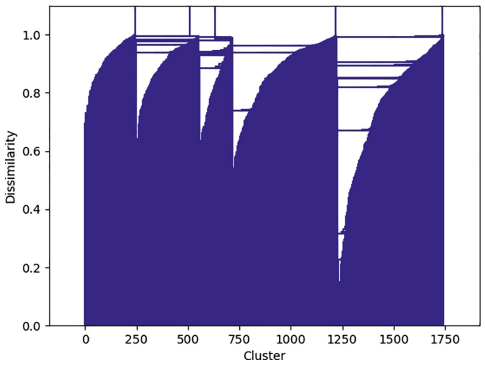

Fig. 2. Dendrogram from party data using an AHC algorithm. Average linkage was used.

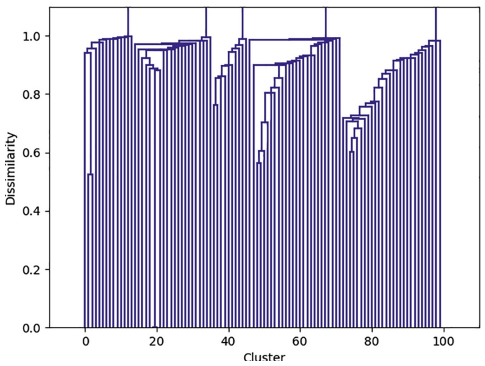

Fig. 3. Dendrogram from party data using a two-stage method

of points in the neighborhood is greater than or equal to L), then x is called a core point. This means that a core point has a enough number of points in its neighborhood. On the other hand, an isolated point is inclined to be a non-core point. The algorithm of DBSCAN starts from a core point and searches another core point in the neighborhood and connects it into the same cluster until no point is connected. The final point connected may not be a core-point. The final point is called a node point.

Let us consider the single linkage clustering in which only core points are clustered. Moreover merging in **AHC** is stopped when the level of merging m_C becomes lower than ϵ, and then the obtained clusters are output. We now have the following proposition.

Proposition 1. *Let the clusters obtained by DBSCAN be $G_1, \ldots, G_K$, and let the clusters obtained by the single linkage (with the stopping parameter ϵ) be $F_1, \ldots, F_K$. Take an arbitrary G_i. Then there is F_j such that $F_j \subseteq G_i$. Moreover if $F_j \neq G_i$, any $x \in G_i - F_j$ is a node point.*

This proposition implies that the result of DBSCAN is similar to clusters obtained from the single linkage for core points. In such a case a non-core point are allocated to a cluster of core points using a simple allocation rule such as the k-nearest neighbor rule.

5 Conclusion

An overview toward new algorithms for clustering categorical and mixed data has been given. Basic methods are reviewed and new methods are shown, which includes a two-stage agglomerative hierarchical algorithm with an example on Twitter and a theoretical results on the relation between DBSCAN and the single linkage.

An important problem of validation of clusters was not discussed, since this problem should be considered in a specific context of a practical application.

To handle a large-scale problem is still difficult in the sense that more efficient algorithms should be developed and also the interpretation problem of clusters should be solved. The latter problem needs knowledge of application domains.

Possible applications of methods herein include not only the categorical and mixed data, but also network clustering such as SNS (Social Networking Services) analysis.

Acknowledgment. This paper is based upon work supported in part by the Air Force Office of Scientific Research/Asian Office of Aerospace Research and Development (AFOSR/AOARD) under award number FA2386-17-1-4046.

References

1. Anderberg, M.R.: Cluster Analysis for Applications. Academic Press, New York (1973)
2. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: Proceedings of SODA 2007, pp. 1027–1035 (2007)
3. Bezdek, J.C.: Fuzzy Mathematics in Pattern Classification, Ph.D. Thesis, Cornell University, Ithaca, NY (1973)
4. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer, Norwell (1981)
5. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exp. P10008 (2008)
6. Davé, R.N.: Characterization and detection of noise in clustering. Pattern Recogn. Lett. **12**, 657–664 (1991)
7. Davé, R.N., Krishnapuram, R.: Robust clustering methods: a unified view. IEEE Trans. Fuzzy Syst. **5**(2), 270–293 (1997)
8. Dunn, J.C.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Cybern. Syst. **3**, 32–57 (1973)
9. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD 1996, pp. 226–231 (1996)
10. Everitt, B.S.: Cluster Analysis, 3rd edn. Arnold, London (1993)
11. Fujiwara, S.: Hierarchical Clustering for Directed Network Data, Master's thesis. University of Tsukuba, Master's Program in Risk Engineering (2017). (in Japanese)
12. Honda, K., Oshio, S., Notsu, A.: Fuzzy co-clustering induced by multinomial mixture model. J. Adv. Comput. Intell. Intell. Inform. **19**(6), 717–726 (2015)
13. Kaizu, Y., Miyamoto, S., Endo, Y.: Hard fuzzy C-Medoids for asymmetric networks. In: Proceedings of 16th World Congress of the International Fuzzy Systems Association (IFSA 2015), 30 June–July 3, Gijon, Spain, pp. 435–440 (2015)
14. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York (1990)
15. MacQueen, J.B.: Some methods of classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1 (University of California Press 1967), pp. 281–297 (1967)
16. Miyahara, S., Miyamoto, S.: A family of algorithms using spectral clustering and DBSCAN. In: Proceedings of 2014 IEEE International Conference on Granular Computing (GrC 2014), Noboribetsu, Hokkaido, Japan, pp. 196–200, 22–24 October 2014
17. Miyamoto, S., Agusta, Y.: An efficient algorithm for ℓ_1 fuzzy c-means and its termination. Control Cybern. **24**(4), 421–436 (1993)
18. Miyamoto, S.: Fuzzy Sets in Information Retrieval and Cluster Analysis. Springer, Heidelberg (1990)
19. Miyamoto, S., Ichihashi, H., Honda, K.: Algorithms for Fuzzy Clustering. Springer, Heidelberg (2008)
20. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)
21. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Phys. Rev. E **69**, 066133 (2004)
22. Tamura, Y., Miyamoto, S.: A method of two stage clustering using agglomerative hierarchical algorithms with one-Pass k-Means++ or k-Median++. In: Proceedings of 2014 IEEE International Conference on Granular Computing (GrC2014), Noboribetsu, Hokkaido, Japan, pp. 281–285, 22–24 October 2014