Clustering Around Medoids based on Ultrametric Properties

S. Fouchal
Laboratoire LaISC
Université de Strasbourg
fouchal@unistra.fr

M. Bui
Laboratoire LaISC
Ecole Pratique des Hautes Etudes
marc.bui@ephe.sorbonne.fr

I. Lavallée
Laboratoire LaISC,
Université Paris 8
ivan.lavallee@univ-paris8.fr

Abstract—The FFUCA (Fast and Flexible Unsupervised Clustering Algorithm) is a fast clustering method based on ultrametric properties. It aggregates data in the same way as partitional methods. But, it elects representatives differently. Indeed, in FFUCA the cluster representatives are deduced from an ultrametric structure built from a sample data. This ultrametric structure gives the data behavior according to used distance. Thus the results are independent from the cluster representatives. We propose in this paper an extension named FFUCAAM to change for better the quality of clusters. Indeed, we improve the election of these representatives. We substitute them by mediods after every new aggregation. This extension increases the complexity in the average case to \(O \left( \sum_{i=1}^{k} m_i^2 \right)\) where \(k\) is the number of the resulting clusters and \(m_i\) is the size of the cluster \(C_i\). In fact, its computational cost is increased but it still less than \(O(n^2)\), thus it remains applicable to large databases.

I. INTRODUCTION

Data mining is an important tool for transforming data to information. It highlights the proximity in a data set. Data mining is used in a large range of profiling practices. Commonly it includes four classes of tasks [1]:

1) Classification: A supervised process which ranks data into fixed number of classes (e.g. Nearest neighbor, Naive Bayes classifier, Neural network).
2) Clustering: An unsupervised process which gathers data in non fixed number of groups. The clustering proceeds in natural way compared to the classification.
3) Regression: Aims to find a modeling function of data with the least error (e.g. Genetic Programming).
4) Association rule learning: Detects relationships between variables.

This paper is focused on the clustering process. Indeed, we propose an extension of the FFUCA (Fast and Flexible Unsupervised Clustering Algorithm) method (cf. Section IV) [2] [3].

Clustering is used in a large variety of fields [4], and many approaches were developed: hierarchical, partitioning, density-based, ...

Therefore, several problems have emerged in this context, such as :

- the feasibility in the large databases which is straight related to the computational cost of methods;
- the dependency of the results on the initial parameters in many approaches (e.g k-means, OPTICS,...) [5];
- detecting clusters which have a concave shape;
- the flexibility regarding the processed context;
- ...

We have proposed in [2] [3] FFUCA method as solution to clustering problems notably to the feasibility in large databases and the flexibility to the studied context. In fact, it is a fast and flexible method which has a computational cost of \(O(n) + \epsilon\) in average case and \(O(n^2) + \epsilon\) in rare worst case. This complexity still the same despite the evolution of the data size. In addition, the user can parameter entries and refine results, that makes easier adapting the method to the context.

We suggest in this paper an extension , based on medoids, of the FFUCA method to highlight the quality of results and to detect better non spherical cluster shapes. Indeed, in FFUCA the cluster representatives are static. In this version, we replace them by dynamic ones. After every aggregation we substitute the representative of the considered cluster by its medoid. Consequently, using dynamic representatives improves the quality of results and the processing of concave shapes. This new method is named FFUCA Around Medoids (FFUCAAM).

This paper is organized as the following: In Section II we present a brief overview of the clustering strategies. In Section III, we give definitions of the important notions used in our method. We recall in Section IV the principle of FFUCA and we present its extension FFUCAAM. Finally, in Section V we give our conclusion and perspectives.

II. RELATED WORKS

The clustering can be summarized by the three following steps [6]:

1) Feature selection and extraction: Preprocessing techniques used, either or both, allow obtaining an appropriate set of features.

2) Pattern proximity: Calculation of similarity (or dissimilarity) between pairs of objects. It is essential to
most clustering procedures.

3) Grouping elements: It can be carried in different way, the most known strategies are defined below [6]:

- **Hierarchical clustering:** Agglomerative (“bottom-up”) (e.g. WPGMA [7], [8], UPGMA [9]) or divisive (“top-down”) (e.g. DIVCLUS-T [10]). It provides a hierarchical tree which is structured but is difficult to interpret when the data set size exceeds a few hundred of elements. The complexity of this kind of algorithms is at least $O(n^2)$ [11]. This complexity makes this strategy infeasible in large databases.

- **Partitional clustering:** Introduced to overcome the problem of complexity. The partitional approach creates clusters by dividing the whole set into $k$ subsets (e.g $k$-means [12], $k$-medoids, PAM [13] ...). The element aggregations to clusters are made around $k$ chosen elements. This process avoids the mutual calculation of similarity. Thus, it decreases the computational cost from $O(n^2)$ to $O(n)$. The major drawback of partitional algorithms is the dependency of the result on the $k$ preselected elements. In general, these methods are sensitive to non spherical shape clusters.

- **Density-based clustering:** Commonly this kind of methods are strong regarding cluster of arbitrary shapes. In this strategy, clusters are regarded as a dense regions leading to the elimination of the noise (e.g. DBSCAN, OPTICS, DENCLUE). The computational cost of these method is generally less than $O(n^2)$ [14], [15], [16], [17], [4].

- **Other approaches:** Many ad-hoc approaches of clustering were developed, such as grid-based methods and graph-based methods,...they are little or rarely used compared to the approaches above cited.

Centroids (e.g $k$-means [12]) and medoids (e.g PAM [13]) were widely used in clustering. They are used in particular in partitional methods to represent clusters in a well manner.

Centroid designates the centered element in a given set. Generally, it does not belong to the considered cluster. Centroids are based on the calculation of the average, thus they are applicable only on numeric data. Their calculation does not affect the general computational cost.

Medoid refers to the prototype of a given set. It is the more similar element to the whole set of elements. The computational cost of determining a medoid is equal to $O(m^2)$ where $m$ is the size of the considered cluster. In spite this high complexity, medoids offer number of advantages: they improve the quality of the results, they across detection of outliers and change for the better detecting the arbitrary shapes [5] [16]. This robustness is important to the flexibility regarding the studied context.

To change for better the quality of FFUCA’s results we suggest in this paper to replace the static representatives of clusters by medoids after every new aggregation. The complexity of FFUCA method is mostly $O(n) + \epsilon$ and seldom (in rare worst case) $O(n^2) + \epsilon$, where $\epsilon$ is equal to $O(s^2)$ and $s$ is the size of a sample data. Note that even if the size of data increases, the complexity of FFUCA, the amortized complexity [18], [19], [20], remains the same [2] [3].

In addition, FFUCA can provide overlapped clusters, (i.e where one element can belong to more than one or more than two clusters, more general than weak-hierarchy), see [15], [16] for detailed definitions.

Using medoids increases the complexity in the average (frequent) case, but still it less than $O(n^2)$. In fact, the new cost is of $O(\sum_{i=1}^{k} n_i^2) + \epsilon$, where $n_i$ is the cardinal of the cluster $C_i$. In the worst case, the complexity still equals to $O(n^2) + \epsilon$ (c.f. Section IV).

III. Definitions

**Definition 1:** A metric space is a set endowed with distance between its elements. It is a particular case of a topological space.

**Definition 2:** We call a distance on a given set $E$, an application $d: E \times E \rightarrow \mathbb{R}^+$ which has the following properties for all $x, y, z \in E$:

1. (Symmetry) $d(x, y) = d(y, x)$,
2. (Positive Definiteness) $d(x, y) \geq 0$, and $d(x, y) = 0$ if and only if $x = y$.
3. (Triangle Inequality) $d(x, z) \leq d(x, y) + d(y, z)$.

**Definition 3:** Let $(E, d)$ be a metric space. If the metric $d$ satisfies the strong triangle inequality:

$$\forall x, y, z \in E, d(x, y) \leq \max\{d(x, z), d(z, y)\}$$

then it is called ultrametric on $E$, and $(E, d)$ is an ultrametric space [21] [22] [23].

**Remark 1:** An ultrametric space is homeomorphic to a subspace of countable product of discrete spaces [7] [21].

**Remark 2:** A topological space is ultrametrizable if and only if it is homeomorphic to a subspace of countable product of discrete spaces [7] [21].

**Remark 3:** A finite set $E$ endowed with a distance $d$ is classifiable if and only if $d$ is an ultrametric distance on $E$ [7] [21].
Definition 4: Let $E$ be a finite set, endowed with a distance $d$, $E$ is classifiable for $d$ if: $\forall \alpha \in \mathbb{R}^+$ the relation on $E$: 

$$\forall x, y \in E, x R_\alpha y \Leftrightarrow d(x, y) \leq \alpha$$

is an equivalent relation.

Thus, we can provide a partition from $E$ as [24]:

- $d(x, y) \leq \alpha \Leftrightarrow x$ and $y$ belong to the same cluster, or,
- $d(x, y) > \alpha \Leftrightarrow x$ and $y$ belong to two distinct clusters.

Example 1: $x$, $y$ and $z$ are three points of plan endowed with an Euclidean distance $d$, we have:

$$d(x, y) = 2, d(y, z) = 3, d(x, z) = 4.$$ 

The set $E$ is not classifiable for $\alpha = 3$. The classification leads to inconsistency.

Remark 4: To reduce the computational complexity of the clustering process, many algorithms build clusters around a specific elements such as centroids, and medoids [5].

Definition 5: Centroid

We name $x^*$ centroid in a given set of elements $E$ every element defined as the following:

$$x^* = \frac{1}{|C_a|} \sum_{x_i \in C_a} x_i$$

Remark 5: In the case of quantitative values $x_i$ (continue or discrete), the centroid is defined by the average value of the cluster elements. It is defined by the center of gravity of the considered cluster. Generally, it centroid do not belong to the cluster elements [16].

Remark 6: When the values of $x_i$ are qualitative (i.e color, form,...), the definition of a centroid do not have a sense since the classic operators are not applicable. Then, we look for the prototype (the more representative element) named “medoid” among cluster elements [16].

Definition 6: Medoid

Let $E$ be the set of the processed element. Consider a cluster $C_a$ of elements on $E$ and $d$ a dissimilarity measure on $E$, the medoid of $C_a$ is $x^* \in C_a$ defined as :

$$x^* = \text{argmin}_{x_i \in C_a} \frac{1}{|C_a|} \sum_{x_j \in C_a} d(x_i, x_j)$$

Remark 7: The medoid of a given cluster is the element which has the minimal average dissimilarity with all other elements of the same cluster (i.e it is the more similar element to all elements of a cluster).

Definition 7: Outliers

They are non frequent observations which are far from dense regions (i.e clusters). We can describe them as observations which were generated by different process [5] [16].

IV. CONSIDERED METHOD FFUCAAM

The FFUCA method is a general method, it can be applied to any kind of data since it uses metric proximity measure. It has tow strong features: it is fast and flexible to the processed data (domain and type).

We propose here an extension of FFUCA based on medoids named FFUCAAM. The aim is to provide a stronger result regarding proximity and to process better non spherical shaped clusters. Indeed, we change the static cluster representatives by dynamic ones. We use medoids to define representatives.

A. Recall: FFUCA principle

The main idea of FFUCA consists in exploiting ultrametric properties to deduce the data behavior according to the used distance. Indeed, we build an ultrametric space from just a
sample data (subset). This sample data is chosen uniformly at random, and has size \( s \) (petty compared to \( n \)) chosen by the user expert of the processed domain. Then, we deduce the data behavior from this ultrametric space. Finally, we aggregate elements according to the deduced information. Considering a data set \( E \) endowed with distance \( d \), the steps of FFUCA are detailed below:

**Assumption 1:** If a part of a \( E \) is classifiable with \( d \) then the whole set \( E \) is classifiable with \( d \).

The FFUCA method is composed of the following steps:

**Step 1:** Choose uniformly at random a sample data from \( E \) (cf. Figure 3);

**Step 2:** Cluster hierarchically the sample data, and represent the distances on the resulting dendrogram, thus the ultrametric space is built (cf. Figure 4);

**Step 3:** Define clusters’ sizes (cf. Figure 5).

**Step 4:** Choose cluster representatives using information of the last step (as the size of the data set is small we use the same data chosen for the sample as representatives).

**Step 5:** Pick the rest of nodes one by one and compare them (according to \( d \)) with the cluster representatives (cf. Figure 6):

- If the compared data is close to one (or more) representative(s) according to thresholds, then aggregate it to the same cluster(s) as this(those) of the closest representative(s) (cf. Figure 7 we see the aggregation of a new element to the closer cluster which is in the top left);
• Else, create a new cluster which will be represented by the remote data (c.f. Figure 8 and Figure 9).

Figure 8. Comparing a far element with representatives

Figure 9. Creation of a new cluster represented by the farest element

We see in the final result (c.f. Figure 10) that FFUCA detect well outliers and generates clusters with different sizes.

Figure 10. Final result including where outliers are well detected

To change for better the quality of the result we propose in the next section an extension of FFUCA based on medoids.

B. Extension: FFUCA Around Medoids (FFUCAAM)

The proposed extension concerns essentially on the election of representatives (i.e step 4) and the aggregation step (i.e step 5). Indeed, instead of using the same cluster representatives in spite changing clusters elements during all the process, we reelect after every new aggregation. The new representative is defined by the mediod of the considered cluster (c.f. Figure 11).

Figure 11. Definition of new representative (medoid)

Using medoids to represent cluster improves the consistency of clusters. Medoid is the more representative elements within a cluster as we can see in the Figure 12. This extension improves also detecting clusters with non spherical forms and outliers [5].

Figure 12. Resulting partition with new representatives

C. Computational complexity

Proposition 1: The computational cost of FFUCAAM, in average case equal to $O(\sum_{i=0}^{k} m_i^2) + \epsilon$ where $m_i$ is the cardinal of the cluster $C_i$ and $k$ is the number of clusters. In the worst case the computational cost still $O(n^2) + \epsilon$.

Proof: In the average case, the complexity is less than $O(n^2)$. During a single passage in the processed data set we calculate the medoid for every cluster. The calculation of one medoid costs $O(m^2)$ where $m$ is the size of the considered cluster. The medoids are calculated locally, thus the general cost is equal to the sum of these calculations. The $\epsilon$ is due to the building of the ultrametric space from the data sample, this operation is detailed in [2] [3].

In the worst case, the algorithm provides only singletons we do not need to calculate medoids.
D. Algorithm

Procedure FFUCAAM

• Variables:
  1) Metric space \( (\text{data set } E \text{ of size } n, \text{distance } d) \);
  2) \( C_n \) is the cardinal of the cluster \( C_i \) and \( k \) is the number of clusters. It still less than \( O(n^2) \).

• Begin:
  1) Choose a sample data;
  2) Cluster hierarchically this sample data and build an ultrametric space;
  3) Compute the complexity of FFUCAAM is, in average the prototypes they enable also processing all kind of data.
  4) For \( i < n - k ; i ++ \):
     a. \( \sum \) calculate \( d(i,j) \):
        * If \( d(i,j) \leq C_j \) size:
          * Then: aggregate \( i \) to \( C_j \);
          * Else: create a new cluster and aggregate \( i \) to this cluster.
        * End If
  5) End for
• End

V. CONCLUSION

Thanks to its linear complexity, the FFUCA is able to process large databases as the majority of partitional methods.
But, the FFUCA uses formal properties to choose cluster representatives. It gets the data behavior before building clusters.
Thus, it provides a more natural partition.

To make more strong the intra-cluster (and inter-clusters) inertia, we proposed in this paper an extension of the FFUCA, the FFUCAAM. Indeed, we represented clusters by medoids.
In spite their elevated computational cost compared to that of centroids, we chose medoids because: in addition to defining the prototypes they enable also processing all kind of data. The computational complexity of FFUCAAM is, in average case equal to \( O(n^2) + \epsilon \). and makes it applicable to large databases. The complexity in the worst case still \( O(n^2) + \epsilon \).

REFERENCES
