New data induced metric for density based clustering

Matthew King and Gleb Beliakov
School of Information technology
Deakin University
221 Burwood Hwy, Burwood 3125, Australia

Abstract
Cluster Analysis is a fundamental data exploration technique in which there exist a variety of algorithms arising in many different areas of the literature. One of the most popular clustering algorithms is fuzzy c-means which minimises the weighted within-group sum of squares functional. However, this algorithm has a major drawback in that it will only find clusters of the same simple convex shape. In reality datasets can contain clusters of a variety shapes, including complex non-convex structures the geometry of which cannot be analytically described. This paper proposes a new data induced metric to be used in the fuzzy c-means functional that results in an algorithm able to find clusters of unknown arbitrary shapes. The proposed metric is based on density information obtained using the Delaunay triangulation. The resulting optimisation problem is now non-smooth, and the alternating optimisation technique used in fuzzy c-means can no longer be applied. Instead we have made use of a recent method of non-smooth optimisation to solve the problem.

1. INTRODUCTION

Exploratory Data Analysis is the process of converting raw data into useful information. One of the most important tasks to perform during this process is Cluster Analysis. Cluster Analysis involves grouping together similar data objects so that we obtain a set of groups satisfying the following two conditions:

(i) homogeneity within a group
- that is, the data objects in the group should be similar to each other.
(ii) heterogeneity between groups
- that is, data objects in different groups should be dissimilar to each other

We assume that each data object can be represented as a point in $d$-dimensional space where $d$ is the number of attributes. Therefore, data objects will now be referred to as data points and thus a dataset is written as: $X = \{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}^d$ where $x_k$ is the $k^{th}$ data point, $N$ is the number of data points, and $\mathbb{R}^d$ is $d$-dimensional Euclidean space. Plots of example two-dimensional datasets are shown below.

![Dataset 1 – 3 convex clusters](image1)
![Dataset 2 – 2 clusters (1 non-convex)](image2)
![Dataset 3 – 3 clusters (2 non-convex)](image3)

Figure 1 – Example Datasets


In two dimensions, a visual inspection can clearly indicate the number and type of clusters present. For instance, Dataset 1 contains clusters of the same simple convex shape, whereas the clusters present in Datasets 2 and 3 include ones of complex non-convex structures.

There are many different clustering algorithms found in the literature arising from varying disciplines such as pattern analysis, data mining, biology, archaeology and geographical analysis. Differing clustering algorithms will produce different results given the same dataset. This is in most part due to the similarity/dissimilarity measure that is used to define the clusters found by the algorithm. For instance, a traditional algorithm that can correctly cluster Dataset 1, such as the popular fuzzy c-means (FCM), will have great difficulties with Datasets 2 and 3. A new approach to clustering is to consider a similarity/dissimilarity metric that is based on the data and thus is specific to that particular dataset. Such an idea would result in an algorithm that allows the data to 'speak for itself' rather than dictating what clusters are to be found. Density-based clustering is an example of such an approach. Algorithms of this type use the dataset to approximate a density function, which is then used to cluster the data.

This paper examines a modification of the FCM algorithm. The traditional FCM algorithm works well when all the clusters are of the same simple convex shape as with Dataset 1, however it fails when complex non-convex clusters are present such as those in Datasets 2 and 3. It forces clusters of the same hyper-elliptical shape regardless of whether this is the true underlying structure or not. The proposed modification incorporates density information into the dissimilarity metric in order to allow FCM to find arbitrarily-shaped non-convex clusters. This is similar to the approach proposed in [1]. However, a new density estimation technique based on the Delaunay triangulation is used, which is expected to be favourable when dealing with small numbers of higher-dimensional data.

This paper is structured as follows: section two briefly looks at the traditional clustering algorithms, section three discusses the density-based clustering approach, section four introduces density estimation based on the Delaunay triangulation, section five outlines the proposed density-based fuzzy c-means algorithm, and section six provides some preliminary results.

2. TRADITIONAL CLUSTERING ALGORITHMS

There are essentially two classical approaches to clustering found in the literature: hierarchical and partitional. Hierarchical clustering generates a hierarchical tree of nested clusters, known as a dendrogram. Each level of the dendrogram corresponds to the situation where the number of clusters is $c (c = 1, 2, \ldots, N)$, where at the top level, or root of the dendrogram, there is one cluster comprising of all the data points and at the bottom level there are $N$ singleton clusters. This approach to clustering is valuable when information as to the structure of the clusters is required rather than a single clustering output. If a single clustering solution is required, then a termination criterion must be specified or a post-clustering visual inspection of the dendrogram must be performed by a domain expert. In either situation subjective judgement is needed.

Hierarchical algorithms can be classified as either agglomerative or divisive, based on how the hierarchical decomposition is constructed. Agglomerative algorithms take a bottom-up approach by starting with $N$ singleton clusters and at each step merging clusters that are ‘close’ to each other. Divisive algorithms on the other hand take a top-down approach and at each step split a cluster into two. The most common hierarchical algorithms are the single-link, complete-link, average-link, and minimum-variance agglomerative algorithms (see [2-5] for details). The algorithms differ in the definition used to measure the similarity/dissimilarity between clusters. The main advantages of the hierarchical approach are the simplicity of the algorithms to understand and implement, and that no prior knowledge of the number of clusters present is required. The disadvantages however are the choice of a termination criterion, and that the
Partitional clustering algorithms produce a single c-partition of the dataset, where c is the pre-defined number of clusters. It is usually found by optimising an objective function that measures the ‘desirability’ of clustering solutions (i.e. ‘desirability’ of different c-partitions). The most common algorithm is the c-means (or k-means) algorithm or its extension to fuzzy clustering, the fuzzy c-means (FCM) algorithm (discussed in [6]). The FCM algorithm minimises the weighted within-group sum of squares (WGSS) functional:

$$J_m = \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik})^m (d_{ik})^2$$

where $U = [u_{ik}]$ is the $c \times N$ fuzzy membership matrix with $u_{ik}$ being the membership of the $k^{th}$ data point in the $i^{th}$ cluster, $u_{ik} \in [0,1]$ , $\sum_{i=1}^{N} u_{ik} = 1 \ \forall k$ , $0 < \sum_{k=1}^{c} u_{ik} < N \ \forall i$ , and $u_{i} \in U$ (i.e. the $i^{th}$ row) corresponds to the $i^{th}$ cluster; $v = (v_1, v_2, ..., v_c) \in \mathbb{R}^{cd}$ is the set of cluster prototypes where $v_i \in \mathbb{R}^d$ is the cluster prototype/centre of the $i^{th}$ cluster $u_i$; $d_{ik} = d(v_i, x_k) = \|v_i - x_k\|$ where $\|$ is any inner product induced norm on $\mathbb{R}^d$; and $m \in [1, \infty)$ is the weighting exponent which controls the ‘fuzziness’ of the solution.

$J_m$ is a complex non-convex function that possesses many local minima. However, $J_m$ as a function of $U$ with $v$ fixed, and $J_m$ as function of $v$ with $U$ fixed are both convex functions. Therefore, FCM minimises $J_m(U,v)$ by iteratively updating $U$ and $v$ using explicit solutions $U^*$ and $v^*$ that satisfy the necessary and sufficient conditions for local minima of $J_m(U)$ and $J_m(v)$ respectively. This method is called Alternating Optimisation (AO). The expressions for $U^*$ and $v^*$ are ([6]):

$$U^* = [u_{ik}^*], \text{ with } u_{ik}^* = \frac{1}{\sum_{j=1}^{c} \left( \frac{d_{ik}}{d_{jk}} \right)^{\frac{m-1}{2}}}$$

and

$$v^* = (v_1^*, v_2^*, ..., v_c^*), \text{ with } v_i^* = \frac{\sum_{k=1}^{N} (u_{ik})^m x_k}{\sum_{k=1}^{N} (u_{ik})^m}$$

[6] shows that convergence of FCM to local minimum of $J_m(U,v)$ is guaranteed (in theory). However, as previously mentioned, $J_m$ possesses many sub-optimal local minima and thus the particular local minimum found by the algorithm is not necessarily the best clustering solution. This consequence, along with the problem of having to specify the number of clusters prior to clustering, are two important disadvantages of FCM and partitional algorithms generally. A third and quite major problem with the FCM algorithm is that only clusters of the same simple convex hyper-elliptical shape can be found. This is because $J_m$ measures the ‘desirability’ of clustering solutions using a global inner product induced norm. [6] and [7] discuss extensions to FCM such as fuzzy c-variates (FCV) and fuzzy c-shells (FCS) that can find clusters of varying structures (eg lines/planes, rings/shells) using definitions of cluster prototypes other

algorithms are computationally expensive when dealing with large datasets. A further disadvantage is that hierarchical algorithms are non-iterative and thus poor assignments cannot be corrected at a later stage.
than the cluster centre point. However, all such extensions can only find clusters of simple regular shapes. Furthermore, the shape must be specified prior to the clustering process.

3. DENSITY-BASED CLUSTERING

All classical clustering algorithms have one thing in common in that dissimilarity between clusters is a general distance-based measure. The properties of such measures dictate the types of clusters found and thus restrict the algorithm’s use to datasets of a certain known structure. Generally nothing is known about the structure of a dataset prior to clustering (especially in higher dimensions) and often the clusters present are of complex non-convex shapes that cannot be analytically described. This results in the need for specific data-induced dissimilarity measures that are based on the structure of the data, rather than a global distance-based metric (eg Euclidean distance) that imposes the same structure on any given dataset. Data density is an obvious property that could be used as such a data-induced dissimilarity/similarity measure. Density-based clustering is a relatively recent approach to the clustering problem that makes use of density information when finding clusters. They rely on the notion that the density of the data should be higher where clusters are present. Intuitively, a high dense region is interpreted as a group of ‘closely’ spaced data points separated by empty space. Mathematically, density can be defined in terms of probability. If we assume that the data set to be clustered $X \subset \mathbb{R}^d$ is a sample taken from a population distribution described by a probability density function (PDF) $f(x)$, then $f(x)$ is defined as:

$$ P(Y \in S) = \int_S f(x) \, dx \quad (4) $$

where $S \subset \mathbb{R}^d$ is a region of d-dimensional space, $P(Y \in S)$ is the probability that a random point $Y \in \mathbb{R}^d$ taken from the population distribution described by $f(x)$ lies in the region $S$, and $f(x)$ gives the point density value at the point $x$.

A cluster, or a dense region, is found in a density-based clustering algorithm by somehow using the point density function $f(x)$. Since the true density function from which the data was sampled is generally not known, an estimate function $\hat{f}(x)$ that approximates the unknown $f(x)$ must be found. Density estimation is a widely studied problem in the area of statistics which can be useful in its own right when exploring data and also when used as part of other data exploration techniques such as clustering. In fact clustering and density estimation are quite similar problems in that both are estimating the true underlying structure of a given dataset. There are two approaches to density estimation: parametric and non-parametric. Parametric techniques assume that the data has been drawn from a known family of distributions (eg Gaussian distribution) and thus approximates the parameters of the known density function (eg $\mu$ and $\sigma$ in the Gaussian distribution function). Clearly such an approach imposes structure on the dataset in the same way done by traditional clustering algorithms like FCM by assuming the clusters belong to certain families of distributions. Non-parametric density estimation is the approach generally taken in density-based clustering since such techniques allow the data to ‘speak for itself’ rather than making rigid assumptions about its structure. The task is to determine a function $\hat{f}(x)$ that estimates a completely unknown density function $f(x)$ using a sample dataset taken from $f$. The most common non-parametric estimation techniques are the histogram, frequency polygons, kernel estimates, k-nearest neighbour estimates and expectation maximisation. For an extensive study of non-parametric techniques, in particular those mentioned above see [8, 9].

Most density-based clustering algorithms found in the literature fall into one of two main classes. The first and most common approach is to use $\hat{f}(x)$ directly to find the clusters by
taking the modes or local maxima of \( \hat{f}(x) \) as the cluster ‘centres’ and the ‘deep enough’ valleys of \( \hat{f}(x) \) as the cluster boundaries. This is usually done using some hill-climbing procedure. Algorithms of this type are proposed in [10-13].

The second class of density-based algorithms is to use \( \hat{f}(x) \) to obtain a data-induced metric based on density information and then to use this metric in a known clustering procedure in replace of the more traditional distance-based measures. The essential difference between a distance-based metric (eg Euclidean distance) and a metric based on \( \hat{f}(x) \) (obtained by a non-parametric estimation technique) is that the latter does not impose any structure on the clusters to be found. One such density-based algorithm is given in [2], where a metric based on the k-nearest neighbours density estimate is used in the single-link hierarchical algorithm. Another algorithm is a modification of FCM proposed in [1] that uses a density approximation based on the histogram to come up with a data-induced metric to be used in place of the norm in the FCM functional. The use of a density-based metric allows clusters of arbitrary non-convex shapes to be found and thus is a substantial improvement on the effectiveness of the traditional FCM algorithm and its extensions. The proposed algorithm in this paper is of this second class of density-based clustering techniques. It is a modification of FCM using a density-based metric in the similar vein as [1]. However, a quite new non-parametric density estimation technique is used that is expected to be more favourable in higher dimensions than the basic histogram measure.

4. THE DELAUNAY TRIANGULATION AND DENSITY ESTIMATION

One-dimensional non-parametric density estimation has been thoroughly studied, and with the optimal choice of a technique’s parameters (particularly the smoothing parameter) adequate results are generally obtained (see [8, 9]). However, one-dimensional cluster analysis is quite trivial, and datasets to be clustered often contain many attributes. The known non-parametric techniques tend to not work well in higher dimensions largely due to the curse of dimensionality ([8]). The sparsity of higher dimensional data results in inadequate estimates unless the number of data points is of a massive size (the number needed grows exponentially with dimension, see [8]).

A new non-parametric density estimation technique that may be able to handle smaller sets of higher dimensional data better than traditional methods is an estimate based on the Delaunay triangulation. Such an idea has been examined in [14, 15]. The Delaunay triangulation is often best described in terms of its dual the Voronoi diagram. Given a set of data points \( X \), its Voronoi diagram \( V(X) \) is a partition of d-dimensional space into \( N \) regions \( v(x_k) \) (\( k=1,2,\ldots,N \)) where \( v(x_k) \) is the set of all locations of the space that are closer to \( x_k \in X \) than any other data point \( x \in X \). The region \( v(x_k) \) is referred to as the Voronoi polygon (in two-dimensional space) or the Voronoi polyhedron (in higher dimensions) associated with data point \( x_k \). The term “closer” is usually measured by the Euclidean distance, however other distances can be used. The Delaunay Triangulation of a dataset \( D(X) \) is obtained by creating an edge between those points that share a \((d-1)\)-dimensional Voronoi face. Provided \( \forall x_k \in X \) is not collinear, \( |X| \geq d+1 \), and exactly \( d+1 \) \((d-1)\)-dimensional Voronoi faces meet at every Voronoi vertex in \( V(X) \), then the Delaunay triangulation exists and is a partition of the convex hull of \( X \), \( CH(X) \), into non-overlapping \( d \)-dimensional simplexes (or in the two-dimensional case triangles, hence its name), where each simplex has \( d+1 \) data points as its vertices. As an example refer to Figure 2, which shows the Delaunay triangulations of the datasets given in Figure 1.
Both the Delaunay triangulation and its dual the Voronoi diagram contain valuable information as to the natural structure of a dataset. There is thus an obvious connection between these two tessellations and the density estimation and clustering problems. In fact, Delaunay triangulations have previously been used directly to cluster data via graph theoretic methods, see [16-19]. For an extensive study of the Delaunay triangulation and its dual the Voronoi diagram see [20].

Given a dataset \( X \subset \mathbb{R}^d \) and a small region of fixed size in \( \mathbb{R}^d \) space, the average density in that region should be proportional to the number of data points \( x_k \in X \) located in that region. Alternatively we can say that the average density of the region should be inversely proportional to the amount of empty space in that region. That is, dense regions of a dataset contain more data points and thus less empty space ‘between’ the points than non-dense regions of the same size. Using this idea we can say that \( \hat{f}(x) \) should be inversely proportional to the amount of empty space surrounding \( x \).

Now, given \( d+1 \) points in \( X \), if the hyper sphere that these points lie on contains no other points in \( X \) we say that these points form an empty hyper sphere. The denser a region is, the smaller all empty hyper spheres formed by points in that region will be. Therefore, \( \hat{f}(x) \) should be inversely proportional to the size of the empty hyper sphere (constructed by \( d+1 \) points in \( X \)) that \( x \) lies inside. Finding the Delaunay triangulation of \( X \) is equivalent to finding all empty hyper spheres of \( X \), where each Delaunay simplex corresponds to an empty hyper sphere. That is, the circumsphere of every Delaunay simplex is an empty hyper sphere and the Delaunay triangulation is a collection of all such possible simplexes constructed from \( d+1 \) data points ([20]). Therefore, the Delaunay triangulation contains topological information as to the amount of empty space between data points. This in turn implies that the Delaunay triangulation is a natural way to estimate the density of a data set. Furthermore, such ‘empty space’/density information is contained in the Delaunay triangulation of any dimension. For this reason it is believed that the Delaunay triangulation may have an advantage over other density estimation methods such as the histogram and kernel estimates when dealing with higher dimensions. Also, a Delaunay triangulation estimate does not require the difficult choice of smoothing parameters.

To come up with a density estimate \( \hat{f}(x) \) using the empty space information contained in the Delaunay triangulation, start with the definition of the density function (4) using regions of the Delaunay triangulation. That is, given a \( d \)-dimensional dataset \( X \subset \mathbb{R}^d \) and its Delaunay triangulation \( D(X) \):

\[
P(Y \in D_i) = \int_{D_i} f(x) \, dx
\]  

(5)
where \(D_i \in D(X)\) (i.e. \(D_i\) is a Delaunay simplex in \(D(X)\)), and \(P( Y \in D_i )\) is the probability that a random point \(Y \in R^d\) taken from the population distribution described by \(f(x)\) lies in the Delaunay simplex \(D_i\), or simply the probability mass of \(D_i\).

Now, assume that the density is constant on each Delaunay simplex and zero outside the convex hull of our dataset. That is, given a Delaunay triangulation \(D(X)\) with \(n_d\) Delaunay simplexes, we restrict \(\hat{f}(x)\) to be the piecewise linear function:

\[
\hat{f}(x) = \begin{cases} 
  c_i & x \in D_i, D_i \in D(X) \\
  0 & x \in CH(X)
\end{cases} \quad i = 1, 2, \ldots, n_d
\]  

(6)

where \(c_i = 1, 2, \ldots, n_d\) is a constant.

Let \(\hat{P}(Y \in D_i)\) be an estimate of \(P(Y \in D_i)\) using density estimate \(\hat{f}(x)\). Then, from (5) and (6) we have:

\[
\hat{P}(Y \in D_i) = \int_{D_i} \hat{f}(x) dx = c_i \int_{D_i} dx = c_i \cdot Vol(D_i)
\]

\[.\]

\[
c_i = \frac{\hat{P}(Y \in D_i)}{Vol(D_i)}
\]

\[.\]

\[
\hat{f}(x) = \begin{cases} 
  \frac{\hat{P}(Y \in D_i)}{Vol(D_i)} & x \in D_i, D_i \in D(X) \\
  0 & x \in CH(X)
\end{cases} \quad i = 1, 2, \ldots, n_d
\]  

(7)

Before investigating the value that should be given to \(\hat{P}(Y \in D_i)\), the consistency of such an estimate should be considered. Let \(S\) be the support of the density function \(f(x)\). Therefore, \(X \subset S\) and as \(N \to \infty\), \(X \to S\). Now, let \(C(D_i)\) be the circumsphere of the Delaunay cell \(D_i \subset D(X)\). Clearly, the volume of empty space in \(S\) approaches zero as \(N \to \infty\). Recall that \(C(D_i)\) is an empty hyper sphere and thus \(Vol(C(D_i)) \to 0\) as \(N \to \infty\). Since \(Vol(C(D_i)) > Vol(D_i)\), it follows that \(Vol(D_i) \to 0\) as \(N \to \infty\).

Now, as \(N \to \infty\), \(D(X) \to S\) (since \(X \to S\) as \(N \to \infty\) and \(D(X) \subset S\)). Also, \(P(Y \in S) = 1\) and \(\hat{P}(Y \in D(X)) = 1 - \hat{P}(Y \notin D(X)) = 1 - \hat{P}(Y \notin CH(X)) = 1 - 0\) (by construction) = 1.

Therefore, \(\lim_{N \to \infty} \hat{P}(Y \in D(X)) = P(Y \in D(X))\). Obviously the consistency of the estimate is dependent on how \(\hat{P}(Y \in D_i)\) is calculated. It is known that as \(N \to \infty\), \(P(Y \in D_i) \to K_i\), where \(K_i\) is the proportion of \(X\) in \(D_i\). Therefore, a consistent estimator can be obtained by estimating the probability mass of \(D_i\) as:

\[
\hat{P}(Y \in D_i) = \frac{\sum x_k}{N} = \frac{n_i}{N} = K_i
\]  

(8)

However, every Delaunay simplex contains exactly \(d+1\) data points located at the vertices of the simplex and these points in general belong to more than one simplex. Therefore, rather than counting the number of data points falling in each cell, as is done with the histogram density estimate, we must come up with an appropriate approximation of the proportion \(K_i\). Assuming
an appropriate approximation of the proportion of data points falling within each Delaunay simplex is known, then \( \hat{f}(x) \) defined in (7) is a consistent estimator since:

\[
\lim_{N \to \infty} \hat{f}(x) = \lim_{N \to \infty} \frac{\hat{P}(Y \in D_i)}{\text{Vol}(D_i)} = \lim_{N \to \infty} \frac{K_i}{\text{Vol}(D_i)} = \lim_{N \to \infty} \frac{P(Y \in D_i)}{\text{Vol}(D_i)} = f(x)
\]

Since there is an equal number of points in each simplex, one obvious solution to the problem of estimating \( K_i \) and the one used in [14] is to assume equal probability mass for each \( D_i \).

Recalling that \( \hat{P}(Y \in D(X)) = 1 \), assuming equal probability mass for each simplex gives:

\[
\hat{P}(Y \in D_i) = \frac{1}{n_d} \quad \forall D_i \in D(X)
\]

(where \( n_d \) = number of Delaunay simplexes).

The idea for a constant \( \hat{P}(Y \in D_i) \) value seems to make sense and in fact [14] showed for the one-dimensional case that \( E(P(Y \in D_i)) = 1/n_d = 1/(N+1) \forall D_i \) (however, the variance is quite high).

Another approach is, for each data point \( x_k \in X \), allocate a proportion of \( x_k \) to each simplex it belongs to. Take a data point \( x_k \in X \) and say it belongs to \( x_k \) simplexes \( D_{(k_1)}, D_{(k_2)}, \ldots, D_{(k_n)} \).

Then for each simplex \( D_{(k_j)} \) (\( j=1,2,\ldots,n \)) we allocate a proportion \( p_{k,D_{(j)}} \) of \( x_k \), where

\[
\sum_{j=1}^{n} p_{k,D_{(j)}} = 1 \quad \forall k \quad \text{(note } D_{(k_j)} \text{ is the } j^{th} \text{ simplex that } x_k \text{ belongs to)}. \text{ Then we have:}
\]

\[
\hat{P}(Y \in D_i) = \frac{\sum_{x_k \in D_i} p_{k,D_{(j)}}}{N} \tag{9}
\]

There are many possible ways that \( x_k \) could be allocated to its neighbouring simplexes. One such approach, similar to that mentioned in [15] is to allocate \( x_k \) in proportion to the volume of its neighbouring simplexes. Another approach is to allocate \( x_k \) evenly to its neighbouring simplexes.

Looking at (7) it can be seen that \( \hat{f}(x) \) is discontinuous at the \( j \)-dimensional faces (\( j=(d-1),(d-2),\ldots,1,0 \)) of every simplex (where 1-dimensional faces and 0-dimensional faces are edges and data points respectively). Furthermore, with the exception of simplexes on the boundary of the \( CH(X) \), these faces have more than one \( \hat{f}(x) \) value. To solve this problem, simply take the density of a \( j \)-dimensional face (\( j=(d-1),(d-2),\ldots,1,0 \)) as the density of the smallest simplex the face is part of. That is, if \( x \in \mathbb{R}^d \) lies on a \( j \)-dimensional facet (\( j=(d-1),(d-2),\ldots,1,0 \)) of \( D(X) \) then:

\[
\hat{f}(x) = \begin{cases} 
\hat{P}(Y \in D_i) & \text{where } D_i = \min_{x \in D_j} \text{Vol}(D_j) \\
\end{cases} \tag{10}
\]

Such a choice seems to work well for clustering since (10) will result in cluster borders having high density values.
5. CLUSTERING AND THE DELAUNAY DENSITY APPROXIMATION

The function \( \hat{f}(x) \) given in (7) with an appropriate definition for \( \hat{P}(Y \in D_i) \) can be used to define a density-based similarity/dissimilarity measure to be used in a conventional clustering algorithm. This paper is considering a metric that can be used in the FCM algorithm in replace of the inner product induced norm. However, it must be noted that such a metric could be applied to any clustering algorithm (eg single-link hierarchical algorithm). Given a dataset \( X \subseteq \mathbb{R}^d \) and points \( u, v \in \mathbb{R}^d \) not necessarily in \( X \) we need to derive a metric \( d(u, v) \). This metric will be used to measure the distance from the \( i \)th cluster centre \( v_i \) to the \( k \)th data point \( x_k \in X \). \( d_{ik} = d(x_k, v_i) \) using \( \hat{f}(x) \) given in (7). The distance will be calculated using graph theoretic methods similar to the ‘data-induced metric’ in [1].

The Delaunay triangulation is a connected graph where the vertices are the data points (0-dimensional faces) and the edges are the simplex edges (1-dimensional faces). This graph can contain explicit density-information obtained from \( \hat{f}(x) \) by converting it into a weighted connected graph where the weights are some function of \( \hat{f}(x) \). We will term this structure the Delaunay density graph \( D_W(X) \). The proposed weight for an edge \( e(x_k, x_l) \) \((x_k, x_l \in X)\) in \( D_W(X) \) is:

\[
W(x_k, x_l) = \frac{1}{\hat{f}(e(x_k, x_l))} \left\| x_k - x_l \right\|_E
\]

where \( \hat{f}(e) \) is the constant \( \hat{f}(x) \) value for the edge \( e \) (note: \( \hat{f}(e) = \hat{f}(x) \), \( x \in e \), \( x \notin x_k, x_l \) calculated using (10)), and \( \left\| \cdot \right\|_E \) is the Euclidean norm.

Such a weight value will result in a metric that takes into account both density information (from \( \hat{f}(x) \)) and distance information (from the Euclidean norm). Note that the reciprocal of the density value is used. This is because we want the weight to reflect the dissimilarity between data points. Therefore, an edge within a cluster (i.e. an edge lying in a dense region of space) will have a small weight due to a large density value and a small edge distance. In contrast, an edge between two clusters will have a small density value and a large edge distance resulting in a large weight.

The distance between any two data points \( x_k, x_l \in X \) can then be taken as the shortest path between \( x_k \) and \( x_l \), where a path is a series of connected edges in \( D_W(X) \). The shortest path calculation is a well known graph theoretic problem that is solved by Dijkstra’s algorithm ([21]). To calculate the distance between points \( u, v \) which do not belong to \( D_W(X) \), we locate the Delaunay simplexes the points belong to, and add extra edges from these points to the vertices of these simplexes, with weights \( W(u, x_k) \) \((k=1,...,d+1)\) and \( W(v, x_l) \) \((l=1,...,d+1)\) computed as in (11) and \( \hat{f}(x) \) defined by (10). Then the distance is also computed as the shortest path between \( u \) and \( v \) along the augmented Delaunay density graph.

It is not difficult to show that \( d(u, v) \) is a metric. Non-negativity and commutativity are trivial. To demonstrate the triangular inequality, observe that for every \( w \), distances \( d(v, w) \) and \( d(w, v) \) are also the shortest paths along the augmented Delaunay density graph. Since \( d(u, v) \) is the shortest distance along the same graph, we obtain \( d(u, v) \leq d(u, w) + d(w, v) \).

This metric will result in small inter-cluster measures since only inter-cluster edges with small weights will be traversed. Similarly, intra-cluster measures will be large due to large-weighted edges lying between clusters having to be crossed to travel from one cluster to the other.
The clustering stage of the new density-based FCM algorithm is thus to minimise the function $J_m$ given in (1) with $d_{ik}$ calculated using $D_w(X)$ as described above. However, the complexity of $J_m$ is now much greater and thus its minimisation is more difficult. The Alternating Optimisation technique can no longer be used to minimise $J_m$ since the expression for $v^*$ given in (3) assumes an inner product induced norm has been used. $J_m$ as function of $v$ with $U$ fixed is now a non-convex and non-smooth function and thus a solution to its minimisation problem can no longer be written down. However, (2) is still the solution to the minimisation of $J_m$ as a function of $U$ with $v$ fixed since its derivation was independent of the $d_{ik}$ values. Therefore, the minimisation of $J_m$ can be solved by minimising (1) (where $d_{ik}$ is the proposed density-based metric) with respect to $v$ using a non-smooth optimisation technique, such as the recent method of discrete gradient proposed in [22], and solving the convex sub-problem $\min\{J_m(U)\}$ fixed every time a function value is needed. That is:

$$\text{Minimise } \{ \hat{\psi}(v) \} \quad (12)$$

where $\hat{\psi}(v) = \min_U J_m(U,v) = \min_U \sum_{k=1}^{N} \sum_{c=1}^{c} (u_{ik})^m (d_{ik})^2 = \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik})^m (d_{ik})^2$.

where $d_{ik}$ is the shortest path between data point $x_k$ and point $v_i$ on the augmented Delaunay density graph and the $u_{ik}$ values are given by (2) (note: the $d_{ik}$ values in (2) are obviously also calculated using $D_w(X)$).

Therefore, the proposed FCM algorithm based on the Delaunay density approximation $\hat{f}(x)$ is as follows:

**Algorithm – Density-Based Fuzzy C-Means**

Input: $X=\{x_1,x_2,...,x_N\} \subset \mathbb{R}^d$, $c$, $m$, initial cluster centres $v^{(0)}=(v_1^{(0)},v_2^{(0)},...,v_c^{(0)})$, and the probability mass function $P(Y \in D)$.  

1. Density Estimation
   - (a) Construct the Delaunay triangulation $D(X)$
   - (b) Using $D(X)$ obtain the density function $\hat{f}(x)$ given in (10)
   - (c) Convert $D(X)$ into the Delaunay density graph $D_w(X)$ by assigning weights given in (11)

2. Clustering
   - Starting with $v^{(0)}$, minimise $\hat{\psi}(v)$ given in (12) using a non-smooth optimisation technique. Each time a function value is needed at the point $v$:
     - (i) Calculate $U$ using (2) with $d_{ik}$ being the proposed density induced metric
     - (ii) $\hat{\psi}(v) = J_m(U,v)$ with $d_{ik}$ being the proposed density induced metric

   Output: $v^*$ found by minimisation procedure
   $U^*$ calculated using (2) with $d_{ik}$ being the proposed density induced metric and $v_i=v_i^*$

**6. PRELIMINARY RESULTS**

The proposed FCM extension has been implemented in C++. The popular general dimension Qhull algorithm ([23]) has been used to construct the Delaunay Triangulation. A non-smooth optimisation technique proposed in [22] that makes use of the discrete gradient is used to minimise (12). As is the case with the traditional FCM, this optimisation technique is local and thus sub-optimal solutions may be found. The search for the global minimum is done by running the optimisation algorithm several times and returning the best solution found. It must be noted that since we are now dealing with a non-smooth optimisation problem, the time taken to converge to a local minimum is significantly greater than that of the FCM algorithm. The algorithm has been tested on three two-dimensional datasets (generated in Maple 9) – one
convex example (Dataset 1), and two non-convex examples (Datasets 2 and 3). Traditional FCM using the Euclidean metric was also run as a comparison. The weighting exponent $m$ was set to 1.5 in both algorithms since the example data sets contained well-separated clusters and thus only a hard partition was required. $\hat{P}(Y \in D)$ was taken to be the constant value $1/n_d$ and both algorithms were run twenty times starting from different random points $V$ to better the chances of obtaining the global minimum.

The proposed algorithm correctly clustered all three datasets, while traditional FCM only worked well with the convex example. The output of both algorithms for the two non-convex examples is shown below (note - each data point is labelled with the cluster number it was found to belong to and the crosses represent the cluster prototypes/centres).

![Figure 3 - Clustering Results](image)

7. CONCLUSION

In this paper we considered the problem identification of clusters of arbitrary shape. Many traditional clustering algorithms are able to find clusters of a given convex shape, imposed by the metric being used. Here we use a density induced metric, which is calculated from the given data. The Delaunay triangulation of the data is used to approximate its density. From this approximation we construct a Delaunay density weighted graph, and define a density induced metric through the shortest path on this graph.

We combined this definition of density induced metric with the well-known fuzzy c-means clustering algorithm. While the density induced metric does not allow us to use alternating optimization, as is usually done in fuzzy c-means, we converted the problem to a non-smooth optimization problem with respect to cluster centres, and applied a recent method of discrete gradient to solve it. Preliminary experiments have produced promising results. However, the importance of the algorithm is expected to be when the dimensionality of the dataset is greater than two, and thus the next step is to investigate the outcome of higher-dimensional examples.
REFERENCES


