Data Induced Metric - Based Algorithm for Detection of Non Convex Patterns

MICHAL KLINESHTERN* and ISAK GATH**

*Dept. of Applied Mathematics and **Dept. of Biomedical Engineering
Technion, Israel Institute of Technology
32000 Haifa
ISRAEL
isak@biomed.technion.ac.il

Abstract: The present study treats the problem of fuzzy clustering of arbitrary shapes. A new approach, based on the principle of Data Induced Metric (DIM) is adopted, distance is defined as medium dependent and the algorithm fits an optimal metric to the given data. A continuous version of the DIM is solved in the present study. A surface is generated by smoothing the data and the distance between any two points is defined by the geodesic on the surface connecting them. The geodesic is calculated by solving numerically the Eikonal equation, using Sethian's Fast Marching method. The DIM principle is implemented to construct a new modified algorithm based on the FKM algorithm. The algorithm has been tested on several data sets, artificial and real, and has been proven to be efficient for partitioning non-convex substructures of arbitrary shapes.

Key-Words: Data Induced Metric; Fuzzy Clustering; Non Convex Patterns; Fast Marching Algorithm; Space and Time Complexities.

1 Introduction

Most objective function cluster analysis algorithms are designed to detect clusters for specific classes of data. The Fuzzy k-means algorithm [1], for instance, is suitable for the detection of hyperspherical clusters, whereas the UOFC and the FKE algorithms [2, 3] solve for the case of hyperelliptoidal and elliptoidal ring shaped clusters, respectively. Most algorithms for handling non-convex clusters have been based on knowledge of the clusters geometrical structure [3, 4]. The present study deals with the problem of partitioning the data into non-convex and well separated arbitrary substructures. A new approach, based on the principle of Data Induced Metric (DIM), is adopted here, and distance is defined as medium dependent. The algorithm fits an optimal metric to the given data. The DI Metric was first introduced by Gath and Iskof [5] in analogy to light propagation in optics. A graph search approach has been suggested for calculating the distance according to the DI Metric. However, graph search algorithms suffer from discretization errors due to metrification error and another problem of this approach is its complexity. An improved version of the MFKM algorithm (Modified Fuzzy K-Means (FKM) algorithm) is developed. A new definition of the DI Metric is introduced. Given a set of n data points, x_i, a surface is generated by smoothing the data points. The distance between two points is the geodesic on the surface connecting them. The geodesic are calculated by solving numerically the Eikonal equation, using Sethian's Fast Marching method [7]. This method yields consistent numerical schemes and guarantees convergence of the solution, with a complexity of O(Nlog(M)), where N is the total number of grid points.
2 The DIM - modified FKM algorithm

The main idea of the DIM is to consider the behavior of the medium. A surface $S$ is built using the objects $x_i$. The geodesic distance is taken as a measure of distance. Using this measure, the trajectory on the surface from a point in one cluster to a point in a different cluster will be longer than the path between two points in the same cluster, i.e. two objects are in the same cluster whenever the path connecting them passes through dense regions (in the surface demonstrated here, pass entirely in the valleys).

**Definition :** Let $S$ be a regular surface. Then according to the Data Induced (DI) metric, the distance $d(x, y)$ between two points $x, y \in S$ is the geodesic from $x$ to $y$ on the surface $S$.

The geodesic satisfies the metric conditions.

3 The Fast Marching Method

The Fast Marching Method, introduced by Sethian [7], is a numerical algorithm for solving the Eikonal equation on a rectangular orthogonal mesh in $O(M \log M)$ steps, where $M$ is the total number of grid points. The technique is based on producing numerically consistent approximation to nonlinear Eikonal equations of the form:

$$\| \nabla T(x, y) \| = F(x, y).$$  \hspace{1cm} (1)

The function, $F(x, y) : R^2 \to R_+$, and the boundary condition that $T$ equals a known function $g(x, y)$ given along a prescribed domain, are typically supplied as known inputs to the equation.

The central idea behind the Fast Marching Methods is to systematically advance the front in an upwind fashion to produce the solution $T$. Thus the information propagates from smaller values of $T$ to larger values. In order to make the algorithm faster the “building zone” is confined to a narrow band around the front. The front is swept ahead in an upwind fashion by considering a set of points in a narrow band around the existing front, proceeding with this narrow band forward, keeping the values of existing points and bringing new ones into the narrow band structure.

Efficient implementation of the Fast Marching Method can be achieved by a fast method for locating the grid point with the smallest value of $T$, based on using a min-heap structure. Given $M$ elements in the heap, this structure allows the change of any element in the heap and reordering of the heap in $O(\log M)$ steps. Thus, the computational efficiency of the total Fast Marching Method for the mesh with $M$ total points is $O(M \log M)$; it takes $M$ steps to touch each mesh point, where each step is $O(\log M)$, because the heap has to be reordered each time the values are changed.

3.1 The new modified algorithm

The new algorithm is obtained from the FKM algorithm by using the DI-metric.

The objective function of the original FKM algorithm is given by:

$$J_q(U, V) = \sum_{j=1}^{N} \sum_{i=1}^{K} (u_{ij})^q d_{ij}^2$$  \hspace{1cm} (2)

where $d_{ij}$ is the distance between the object $x_j$ and the cluster center $v_i$. In the modification, the DI metric is used as a distance measure. $u_{ij}$ is the degree of membership of $x_j$ in cluster $i$ and $q$ is a weighting exponent.

The original FKM algorithm can start with initialization of the membership matrix, $U$, or from initialization of the cluster centers, $V$. Those values are chosen randomly. In the modified algorithm the choice of initial values might influence results of the clustering. In order to improve the results of this algorithm the rule for choosing the initial conditions is as follow:
• The initial conditions are the cluster centers $V^0$.

• The center of a cluster must lie "inside" the cluster, in a high density region. It has to be one of the objects that belong to this cluster (the "closest" object in an Euclidean measure to the calculated center). Thus, the geodesic, calculated according to the DI-$\text{metric}$, from an object in that cluster to this center will pass only inside the cluster (in the valley).

• The centers $V^0$ should belong to different clusters. It is difficult to check if the centers do belong to different clusters, but if they are "far enough" (according to the DI-$\text{metric}$) from one another it is reasonable to assume that they are in different clusters. "Far enough" is an empirical value. It was taken as the average distance from the center to all the objects.

The first center $v^0_1$ is chosen as one of the data objects in the grid cell with the highest density. Then the distance (geodesic) from this center to all other objects is computed and the "average distance" is calculated. The next center is the object which is "far enough" from $v^0_1$. If there are no objects satisfying this requirement, the threshold defining "far enough" is decreased until such an object is found. Each next center must be "far enough" from all the already existing centers. When all initial centers are determined, initialization is completed.

The Modified FKM algorithm is presented below:

1. Initialization:
   • fix $K$ (the number of clusters).
   • choose $q$ from the range $1 < q \leq \infty$ (usually $q$ is equal 2).
   • prepare the grid and generate a surface.

   • choose initial cluster centers $v^0$.

   In the modified algorithm the $coef$ (grid partition parameter), and $a$ (an empty cell value) have to be determined. The default values of these parameters were empirically chosen: $coef = 1.0, a = 0.01$.

2. Compute the degree of membership of all feature vectors in all the clusters:

   $$u_{ij} = 1 / \sum_{k=1}^{K} \left( \frac{d(x_j, v_i)}{q(x_j, v_k)} \right)^{\frac{2}{q-1}}, \forall i, j : 1 \leq i \leq K, 1 \leq j \leq N$$

   For any object $x_j$ which is in the same grid cell with the center $v_r$, the distance $d_{rl}$ is equal to zero. The values of the membership is then: $u_{ij} = 1$ for $i = r, j = l$ and $u_{ij} = 0$ otherwise.

3. Compute new cluster centers according to:

   $$\bar{v}_i = \frac{\sum_{j=1}^{N} u_{ij} x_j}{\sum_{j=1}^{N} u_{ij}}$$

   The new centers do not necessarily lie inside the clusters. The algorithm search for the nearest object $x_j$ that belong to the cluster $i$ (the "nearest" object in an Euclidean measure to the calculated center $v_i$).

4. If $\|V - \bar{V}\| > \varepsilon$ then goto step 2.
   where $\varepsilon$ is a termination criterion between 0 and 1, and $\| \cdot \|$ is some norm.

5. end.

The membership matrix, $U$, presents the result of the algorithm. To obtain a hard partition we can calculate:

$$u^*_{ij} = \max_i u_{ij}$$

where $1 \leq j \leq N, 1 \leq i \leq K$.

Then set $u_{ij} = 1$ for the index $i$ for which
the maximum is obtained, and all other degrees of membership of the \( j \)th object will be equal to zero.

4 Performance of the algorithm

A large number of simulations of non-convex clusters of arbitrary shape has been carried out in order to test the new DIM - modified FKM. Several sets of real data were also investigated.

For comparison, some of the examples were also run using the older version of the DIM - modified FKM [5]. The main differences were the running times and the distance maps.

4.1 Synthetic data

The data in Figure 1 consists of 227 objects, forming a picture of a flower pot (drawn by hand).

![Figure 1: Grid generated for the data (a flower pot). 17 × 17 grid cells.](image)

The number of clusters is \( K = 3 \). The grid generation parameter is \( \eta = 1.2 \).

The grid resolution with this \( \eta \) is \( 17 \times 17 \) see Figure 1.

All other parameters are set to the defaults.

![Figure 2: Surface generated from the data density obtained with three iterations and with no errors. Running time is 6 seconds.](image)

The surface generated by the object densities is shown in Figure 2. There are three separate valleys.

![Figure 3: Final partition of the data](image)

Figure 3 presents the final partition of the data into three clusters. These results are obtained with three iterations and with no errors. Running time is 6 seconds.

If we apply calculation of the DI metric as in [5] the results are the same but the number of flops for finding the distances is 535931 vs. 54643 with the new version of the algorithm.

4.2 Real data

The source image shown in Figure 4 is an image of 10 chromosomes that contains 201 × 105 pixels.

Matlab’s standard procedure for edge detection is applied and the image is converted into a set of points in the plane, containing 1002 data points.

The parameters for this example are:

- \( K = 9 \): The number of clusters is nine.
- \( \eta = 1.55 \): The grid generation parameter. (high resolution).
- \( a = 0.00001 \): An empty cell gets a higher value in the generated surface.
The resolution is higher than the default leading to a better final partition. The grid resolution with this $\eta$ is $49 \times 49$. All other parameters are set to the default values.

Figure 5: Surface generated by the data density

From the grid cell densities a surface is created and the geodesic on it are found (Figure 5). It is easy to see that the clusters (the dense areas in the grid) are valleys.

The algorithm succeeds to detect nine clusters (Figure 6). Running time is 148 seconds. The relatively long running time is due to the large number of objects and to the desired high resolution.

The distance contours around one of the centers in the initial partition are shown in Figure 7. The clusters can be easily distinguished in this figure.

5 Conclusions

The main idea of the present study is to use a Data Induced (DI) Metric in order to detect non-convex well separated clusters in the data structure. The use of the metric has to be efficient and with less discretization error caused by the grid partition.

The new DIM modified Fuzzy K Means algorithm demonstrates an application of the DI Metric for clustering data sets of non-convex clusters of arbitrary shapes.

The algorithm was tested on different types of data: non-convex separated clouds, S-shaped curves with compact body, circular ring surrounding a circle, two concentric approximately elliptical contours, etc. The results demonstrate good performance of the algorithm. The original FKM works with small complexity, but the examples that were taken (non-convex well separate clusters) are un-
solved by it. For the case of overlapping structures, the new modified FKM has no advantages compared to [2, 3, 4]. However, when the clusters are non-convex and well separated it has a clear advantage. The application of the DI Metric to the FKM clustering algorithm extends its capabilities.

The computing time required by the new DIM - modified FKM is acceptable but usually larger than that of the original FKM, yet smaller then that of the modified FKM in [5] (using the same metric). The main difference is the time it takes for calculating the distances. Sethian’s Fast Marching method [7] was used for the distance calculation. This method provides an approach that directly approximates the solution of the underlying partial differential equation through consistent numerical approximations. The speed of the algorithm comes from a heap-sort technique to efficiently locate the smallest element set. Thus, if there are \( M \) total grid cells there is a computational complexity of \( O(M) \) to reach each grid cell. Since the heap can be re-ordered in \( O(\log(M)) \) steps, the computational complexity ends with \( O(M\log(M)) \).

Other approaches for characterization of non-convex patterns of arbitrary shape come under the category of hierarchical clustering, and include graph theoretic methods [8] and mutual nearest neighborhood [6]. The main difference of both methods from the method of clustering developed in the present work is the choice of distance measure. Both methods are based on a predetermined given metric (very often the Euclidean distance measure), whereas in the present work the principle of the DI Metric is used.

References


