A sequential Bayesian alternative to the classical parallel fuzzy clustering model

Arash Abadpour

Imaging Group, Epson Edge, Epson Canada Limited, Canada

ARTICLE INFO

Article history:
Received 8 October 2014
Received in revised form 16 April 2015
Accepted 3 May 2015
Available online 12 May 2015

Keywords:
Data clustering
Fuzzy clustering
Possibilistic clustering
Robust clustering
Bayesian modeling
Sequential clustering

ABSTRACT

Unsupervised separation of a group of datums of a particular type, into clusters which are homogenous within a problem class-specific context, is a classical research problem which is still actively visited. Since the 1960s, the research community has converged into a class of clustering algorithms, which utilizes concepts such as fuzzy/probabilistic membership as well as possibilistic and credibilistic degrees. In spite of the differences in the formalizations and approaches to loss assessment in different algorithms, a significant majority of the works in the literature utilize the sum of datum-to-cluster distances for all datums and all clusters. In essence, this double summation is the basis on which additional features such as outlier rejection and robustification are built. In this work, we revisit this classical concept and suggest an alternative clustering model in which clusters function on datums sequentially. We exhibit that the notion of being an outlier emerges within the mathematical model developed in this paper. Then, we provide a generic loss model in the new framework. In fact, this model is independent of any particular datum or cluster models and utilizes a robust loss function. An important aspect of this work is that the modeling is entirely based on a Bayesian inference framework and that we avoid any notion of engineering terms based on heuristics or intuitions. We then develop a solution strategy which functions within an Alternating Optimization pipeline.

1. Introduction

Many signal and image processing applications require the unsupervised grouping of a set of datums of a particular model into a number of homogenous clusters. While the notion of homogeneity in these applications is defined within the context of the underlying physical phenomena, it is highly advantageous to be able to utilize abstract datum and cluster models and thus to arrive at a class-independent clustering algorithm.

In effect, any clustering algorithm is based on a number of models, some of which may seem trivial choices. Nevertheless, these structural elements dictate the behavior of the respective algorithms. For example, among the popular assumptions, within the field of data clustering, is the notion that datums can be reduced to multi-dimensional vectors and that clusters can be represented as prototypical datums. The Euclidean distance is a distance function of choice in such efforts. We argue that some of the major challenges in the field are direct results of these assumptions and therefore we suggest a critical perspective on de facto accepted aspects of popular data clustering algorithms.

E-mail address: arash@abadpour.com

http://dx.doi.org/10.1016/j.ins.2015.05.007
0020-0255/© 2015 Elsevier Inc. All rights reserved.
Argumenty, intuition and heuristics have been a driving force behind many major developments in the field of data clustering. Under such regimes, objective functions, constraints, and parameters are engineered based on a verbal description of certain perceptual notions, such as “tension”, “force”, and “seizure”. This has led to different formalizations of the clustering problem which are often governed by regularization coefficients. As will be discussed in Section 2, these parameters often have to be set by a user and, generally, setting them improperly has severe implications for the performance of the algorithms. We argue in this paper that a revisit of the construction methodology utilized by data clustering problems can alleviate this challenge.

In this paper, we follow a Bayesian inference framework and derive a loss model for a generic clustering problem. The framework developed in this paper replaces the well-known sum of datum-to-cluster distances with a sequential pipeline within which a notion of outlier rejection emerges. The model developed in this paper is independent of the datum and cluster models applicable to any particular problem class and employs robustification means. We then provide a solution strategy for the developed clustering algorithm.

The rest of this paper is organized as follows. First, in Section 2, we review the related literature and then, in Section 3, we present the developed method. Subsequently, in Section 4, we provide experimental results produced by the developed method on three different problem classes and, in Section 5, we present the concluding remarks.

2. Literature review

2.1. Notion of membership

The notion of membership is a key point of distinction between different clustering schemes. Essentially, membership may be Hard or Fuzzy. Within the context of hard membership, each datum belongs to one cluster and is different from all other clusters. The fuzzy membership regime, however, maintains that each datum in fact belongs to all clusters, with the stipulation that the degree of membership to different clusters is different. K-means [69] and Hard C-means (HCM) [33] clustering algorithms, for example, utilize hard membership values. The reader is referred to [49] and the references therein for a history of K-means clustering and other methods closely related to it. Iterative Self-Organizing Data Clustering (ISODATA) [3] is a hard clustering algorithm as well.

With the introduction of Fuzzy Theory [102], many researchers incorporated this more natural notion into clustering algorithms [4]. The premise for employing a fuzzy clustering algorithm is that fuzzy membership is more applicable in practical settings, where generally no distinct line of separation between clusters is present. Additionally, from a practical perspective, it is observed that hard clustering techniques are extremely more prone to falling into local minima [23]. The reader is referred to [9] for the wide array of fuzzy clustering methods developed in the past few decades.

Initial work on fuzzy clustering was done by Ruspini [86] and Dunn [24] and it was then generalized by Bezdek [9] into Fuzzy C-means (FCM). In FCM, datums, which are denoted as $x_1, \ldots, x_N$, belong to $\mathbb{R}^k$ and clusters, which are identified as $\psi_1, \ldots, \psi_C$, are represented as points in $\mathbb{R}^k$. FCM makes the assumption that the number of clusters, $C$, is known through a separate process or expert opinion and minimizes the following objective function,

$$A = \sum_{c=1}^{C} \sum_{n=1}^{N} w_{nc} \| x_n - \psi_c \|^2.$$  \hspace{1cm} (1)

This objective function is heuristically suggested to result in appropriate clustering results and is constrained by,

$$\sum_{c=1}^{C} f_{nc} = 1, \quad \forall n.$$  \hspace{1cm} (2)

Here, $f_{nc} \in [0, 1]$ denotes the membership of datum $x_n$ to cluster $\psi_c$.

In (1), $m > 1$ is the fuzzifier (also called weighing exponent and fuzziness). The optimal choice for the value of the fuzzifier is a debated matter [64] and is suggested to be “an open question” [101]. Bezdek [91] suggests that $1 < m < 5$ is a proper range and utilizes $m = 2$. The use of $m = 2$ is suggested by Dunn [24] in his early work on the topic as well and also by Frigui and Krishnapuram [29], among others [54]. Bezdek [8] provided physical evidence for the choice of $m = 2$ and Pal and Bezdek [77] suggested that the best choice for $m$ is probably in the interval $[1.5, 2.5]$. Yu et al. [101] argue that the choices for the value of $m$ are mainly empirical and lack a theoretical basis. They worked on providing such a basis and suggested that “a proper $m$ depends on the data set itself” [101].

Recently, Zhou et al. [106] proposed a method for determining the optimal value of $m$ in the context of FCM. They employed four Cluster Validity Index (CVI) models and utilized repeated clustering for $m \in [1.1, 5]$ on four synthetic data sets as well as four real data sets adopted from the UCI Machine Learning Repository [68] (refer to [10] for a review of CVIs and [87] for coverage in the context of relational clustering). The range for $m$ in that work is based on previous research [76] which provided lower and upper bounds on $m$. The investigation carried in [106] yields that $m = 2.5$ and $m = 3$ are optimal in many cases and that $m = 2$ may in fact not be appropriate for an arbitrary set of datums. This result is in line with other works which demonstrate that larger values of $m$ provide more robustness against noise and the outliers. Nevertheless,
significantly large values of $m$ are known to push the convergence towards the sample mean, in the context of Euclidean clustering [101]. Wu [96] analyzes FCM and some of its variants in the context of robustness and recommends $m = 4$.

Rouseeuw et al. [85] suggested replacing $f_{mc}^m$ with $zf_{mc} + (1 - z)f_{mc}^2$, for a known $0 < z < 1$. Klawonn and Hoppner suggested to generalize this effort and to replace $f_{mc}^m$ with an increasing and differentiable function $g(f_{mc})$ [53,52].

Pédrez [80,82,81] suggested to modify (2) in favor of customized $\sum f_{mc}$ constraints for different values of $n$. That technique allows for the inclusion of a priori information into the clustering scheme and is addressed as Conditional Fuzzy C-means (CFCM). The same modification is carried out in Credibilistic Fuzzy C-Means (CFCM) [14], where, the "credibility" of datums is defined based on the distance between datums and clusters. Therefore, in that approach, (2) is modified in order to deflate the membership of outliers to the set of clusters (also see [99]). Customization of (2) is also carried out in Cluster Size Insensitive FCM (csiFCM) [74] in order to moderate the impact of datums in larger clusters on an smaller adjacent cluster. Leski [64] provides a generalized version of this approach in which $\sum \beta f_{mc}^z$ is constrained.

### 2.2. Prototype-based clustering

It is a common assumption that the notion of homogeneity depends on datum-to-datum distances. This assumption is made implicitly when clusters are modeled as prototypical datums, also called clustroids or cluster centroids, as in FCM, for example. A prominent choice in these works is the use of the Euclidean distance function [61]. For example, the potential function approach considers datums as energy sources scattered in a multi-dimensional space and seeks peak values in the field [98] (also see [7,83]). We argue, however, that the distance between the datums may not be either defined or meaningful and that what the clustering algorithm is to accomplish is the minimization of datum-to-cluster distances. For example, when datums are to be clustered into certain lower-dimensional subspaces, as is the case with Fuzzy C-Varieties (FCV) [65], the Euclidean distance between the datums is irrelevant.

Nevertheless, prototype-based clustering does not necessarily require explicitly present prototypes. For example, in kernel-based clustering, it is assumed that a non-Euclidean distance can be defined between any two datums. The clustering algorithm then functions based on an FCM-style objective function and produces clustroids which are defined in the same feature space as the datums [92]. These cluster prototypes may not be explicitly represented in the datum space, but, nevertheless, they share the same mathematical model as the datums [97] (the reader is referred to a review of Kernel FCM (KFCM) and Multiple-Kernel FCM (MKFCM) in [12] and several variants of KFCM in [13]). Another example for an intrinsically prototype-based clustering approach in which the prototypes are not explicitly "visible" is the Fuzzy PCA-guided Robust k-means (FPR k-means) clustering algorithm [43] in which a centroid-less formulation [103] is adopted which, nevertheless, defines homogeneity as datum-to-data proximity. Relational clustering approaches constitute another class of algorithms which are intrinsically based on datum-to-datum distances (for example refer to Relational FCM (RFCM) [40] and its non-Euclidean extension Nerf C-means [37]). The goal of this class of algorithms is to group datums into self-similar bunches. Another algorithm in which the presence of prototypes may be less evident is Multiple Prototype Fuzzy Clustering Model (FCMP) [72], in which datums are described as a linear combination of a set of prototypes, which are, nevertheless, members of the same $R^k$ as the datums are. Additionally, some researchers utilize $L_r$-norms, for $r \neq 2$ [47,11,38,39], or other datum-to-datum distance functions [50].

We argue that a successful departure from the assumption of prototypical clustering is achieved when clusters and datums have different mathematical models. For example, the Gustafson–Kessel algorithm [35] models a cluster as a pair of a point and a covariance matrix and utilizes the Mahalanobis distance between datums and clusters (also see the Gath–Geva algorithm [32]). Fuzzy shell clustering algorithms [54] utilize more generic geometrical structures. For example, the FCV [65] algorithm can detect lines, planes, and other hyper-planar forms, the Fuzzy C Ellipsoidal Shells (FCES) [28] algorithm searches for ellipses, ellipsoids, and hyperellipsoids, and the Fuzzy C Quadric Shells (FCQS) [54] and its variants seek quadric and hyperquadric clusters (also see Fuzzy C Plano-Quadric Shells (FCPQS) [58]).

### 2.3. Robustification

Dave and Krishnapuram [21] argue that the concept of membership function in FCM and the concept of weight function in robust statistics are related. Based on this perspective, it is argued that the classical FCM in fact provides an indirect means for attempting robustness. Nevertheless, it is known that FCM and other least square methods are highly sensitive to noise [14]. Hence, there has been ongoing research on the possible modifications of FCM in order to provide a (more) robust clustering algorithm [63,26]. Dave and Krishnapuram [21] provide an extensive list of relevant works and outline the intrinsic similarities within a unified view (also see [34,18]).

The first attempt to robustifying FCM, based on one account [21], is the Ohashi Algorithm [34,75]. That work adds a noise class to FCM and writes the robustified objective function as,

$$D = \sum_{c=1}^{C} \sum_{n=1}^{N} f_{mc}^m ||x_n - \psi_c||^2 + (1 - z) \sum_{n=1}^{N} \left( 1 - \sum_{c=1}^{C} f_{mc} \right)^m.$$  \hspace{1cm} (3)

The transformation from (1) to (3) was suggested independently by Dave [18,19] when he developed the Noise Clustering (NC) algorithm as well. The core idea in NC is that there exists one additional imaginary prototype which is at a fixed
distance from all datums and represents noise. That approach is similar to modeling approaches which perform consecutive identification and deletion of one cluster at a time [48,107]. Those methods, however, are expensive to carry out and require reliable cluster validity measures.

Krishnapuram and Keller [59] extended the idea behind NC and developed the Possibilistic C-means (PCM) algorithm by rewriting the objective function as,

\[
A = \sum_{c=1}^{C} \sum_{n=1}^{N} t_{nc}^m \|x_n - \psi_c\|^2 + \sum_{c=1}^{C} \eta_c \sum_{n=1}^{N} (1 - t_{nc})^m.
\]

Here, \(t_{nc}\) denotes the degree of representativeness or typicality of datum \(x_n\) to cluster \(\psi_c\) (also addressed as possibilistic degree in contrast to the probabilistic model utilized in FCM). As expected from the modification in the way \(t_{nc}\) is defined, compared to that of \(f_{nc}\), PCM removes the sum of one constraint, shown in (2), and in effect extends the idea of one noise class in NC into C noise classes. In other words, PCM could be considered as the parallel execution of C independent NC algorithms, each seeking a cluster. Therefore, the value of \(C\) is somewhat arbitrary in PCM [21]. For this reason, PCM has been called a mode-seeking algorithm where \(C\) is the upper bound on the number of modes.

We argue that the interlocking mechanism present in FCM, i.e. (2), is valuable in that, not only clusters seek homogenous sets, but that they are also forced into more optimal “positions” through forces applied by competing clusters. In other words, borrowing the language used in [61], in FCM clusters “seize” datums and it is disadvantageous for multiple clusters to claim high membership to the same datum. There is no phenomenon, however, in NC and PCM which corresponds to this internal factor. Additionally, it is likely that PCM clusters coincide and/or leave out portions of the data unclustered [5]. In fact, it is argued that the fact that at least some of the clusters generated through PCM are nonidentical is because PCM gets trapped into local minimum [90]. PCM is also known to be more sensitive to initialization than other algorithms in its class [61].

It has been argued that both concepts of possibilistic degrees and membership values have positive contributions to the purpose of clustering [17]. Hence, Pal et al. [78] combined FCM and PCM and rewrote the optimization function of Fuzzy Possibilistic C-Means (FPCM) as minimizing,

\[
A = \sum_{c=1}^{C} \sum_{n=1}^{N} (f_{nc}^m + t_{nc}^m) \|x_n - \psi_c\|^2,
\]

subject to (2) and \(\sum_{n=1}^{N} t_{nc} = 1\), \(\forall c\). That approach was later shown to suffer from different scales for \(f_{nc}\) and \(t_{nc}\) values, especially when \(N \gg C\) and, therefore, additional linear coefficients and a PCM-style term were introduced to the objective function [79] (also see [94] for another variant). It has been argued that the resulting objective function employs four correlated parameters and that the optimal choice for them for a particular problem instance may not be trivial [61]. Additionally, in the new combined form, \(f_{nc}\) cannot necessarily be interpreted as a membership value [61].

Weight modeling is an alternative robustification technique and is exemplified in the algorithm developed by Keller [51], where the objective function is rewritten as,

\[
A = \sum_{c=1}^{C} \sum_{n=1}^{N} t_{nc} \omega_n \|x_n - \psi_c\|^2,
\]

subject to \(\sum_{n=1}^{N} \omega_n = \omega\). Here, the values of \(\omega_n\) are updated during the process as well.

Frigui and Krishnapuram [29] included a robust loss function in the objective function of FCM and developed Robust C-Prototypes (RCP),

\[
A = \sum_{c=1}^{C} \sum_{n=1}^{N} t_{nc} u_c(\|x_n - \psi_c\|),
\]

Here, \(u_c(\cdot)\) is the robust loss function for cluster \(c\). They further extended RCP and developed an unsupervised version of RCP, nicknamed URCP [29]. Wu and Yang [97] used \(u_c(x) = 1 - e^{-\alpha x}^2\) and developed Alternative HCM (AHCM) and Alternative FCM (AFCM) algorithms (also see [104]).

2.4. Number of clusters

The classical FCM and PCM, and many of their variants, are based on the assumption that the number of clusters is known (the reader is referred to [9, Chapter 4] and [46, Chapter 4] for reviews of this topic). While PCM-style formulations may appear to relax this requirement, the corresponding modification is carried out at the cost of yielding an ill-posed optimization problem [61]. Hence, provisions have been added to existing clustering algorithms in order to address this challenge. Repeating the clustering procedure for different numbers of clusters [32,56,57] and Progressive Clustering are two of the approaches devised in the literature.

Among the many variants of Progressive Clustering are methods which start with a significantly large number of clusters and freeze “good” clusters [56,60,55,58], approaches which combine compatible clusters [20,56,60,55,58,29], and the
technique of searching for one “good” cluster at a time until no more is found [48, 55, 88]. Use of regularization terms in order to push the clustering results towards the “appropriate” number of clusters is another approach taken in the literature [30]. These regularization terms, however, generally involve additional parameters which are to be set carefully, and potentially per problem instance (for example see the mixed C-means clustering model proposed in [78]).

Dave and Krishnapuram conclude in their 1997 paper that the solution to the general problem of robust clustering when the number of clusters is unknown is “elusive” and that the techniques available in the literature each have their limitations [21]. In this paper, we acknowledge that the problem of determining the appropriate number of clusters is hard to solve and even hard to formalize. Additionally, we argue that this challenge is equally applicable to many clustering problems independent of the particular clustering model utilized in the algorithms. Therefore, we designate this challenge as being outside the scope of this contribution and assume that either the appropriate number of clusters is known or that an exogenous means of cluster pruning is available which can be utilized within the context of the algorithm developed in this paper.

2.5. Weighted clustering

Many fuzzy and possibilistic clustering algorithms make the assumption that the datums are equally important. Weighted fuzzy clustering, however, works on inputs datums which have an associated positive weight [48]. This notion can be considered as a marginal case of clustering fuzzy data [25]. Examples for this setting include clustering of a weighted set, clustering of sampled data, clustering in the presence of multiple classes of datums with different priorities [1], and a measure used in order to speed up the execution through data reduction [89, 41, 100]. Nock and Nielsen [73] formalize the case in which weights are manipulated in order to move the clustering results towards datums which are harder to include regularly. Note that the extension of FCM on weighted sets has been developed under different names, including Density-Weighted FCM (WFCM) [41], Fuzzy Weighted C-means (FWCM) [67], and New Weighted FCM (NW-FCM) [45].

2.6. Sequential clustering

FCM, PCM, and the majority of other available clustering techniques, expose all datums to all clusters simultaneously. This is in effect the results of loss models such as (1), in which NC datum-to-cluster distance terms are combined directly. We argue, however, that there are advantages in constructing a sequential clustering framework.

A first approach to restructuring the operation of the clustering algorithm into a sequence, is the technique of detecting one cluster at a time and then removing the corresponding datums from the set. More elaborate works in this category avoid deletion of clustered datums and utilize terms which disincentivize the consideration of previously clustered datums for new clusters. For example, in [93], the authors detect the clusters sequentially and then utilize a regularization term which pushes new clusters off the territory of previous ones. The employed regularization term, however, depends on a set of \( \beta_i \) values and the authors state that “since we do not have an established method for settings \( \beta_i \)’s, these parameters must be set empirically”. As discussed before, we argue that the dependence of the algorithm on such configuration parameters is a significant practical challenge for the use of an algorithm in the context of unsupervised operation.

A second class of sequential clustering approaches serializes the datums and allows the clusters to respond to them in a sequence. Fuzzy Competitive Learning (FCL) [15] and its predecessor Enhanced Sequential Fuzzy Clustering (ESFC) [105] are sample algorithms in this category. This approach is either adopted as a measure to avoid the entrapment of the clusters in local minima, or, in response to the characteristics of problems classes in which datums are in fact discovered sequentially and are not available to the algorithm as a full set.

3. Developed method

In this section, we develop a novel alternative to the generic clustering model utilized in FCM, PCM, and a significant number of other fuzzy, possibilistic, and credibilistic algorithms available in the literature. We show that the hegemony of the classical model, which calculates the sum total of datum-to-cluster distances, can be broken by a novel clustering model which also provides better robustification in addition to an emergent outlier rejection property.

3.1. Model preliminaries

We assume that a problem class is given, within the context of which a datum model is known and denote a datum as \( x \). We also assume that a particular cluster model is provided, which complies with the notion of homogeneity relevant to the problem class at hand, and denote a cluster as \( \psi \).

In this work, we utilize a weighted set of datums, defined as,

\[
\mathbf{X} = \{(\omega_n; x_n)\}, \quad n = 1, \ldots, N, \quad \omega_n > 0,
\]

and we define the weight of \( \mathbf{X} \) as,

\[
\Omega = \sum_{n=1}^{N} \omega_n.
\]
In this context, when estimating loss, we treat $X$ as a set of realizations of the random variable $x$ and write, 

$$p(x_n) = \frac{\omega_n}{\Omega}.$$  

(10)

We assume that the real-valued positive distance function $\phi(x, \psi)$ is defined. Through this abstraction, we allow for any applicable distance function and therefore avoid the dependence of the underlying algorithm on the Euclidean or other specific notions of distance. As examples, when the datums belong to $\mathbb{R}^k$, the Euclidean Distance, any $L_r$ norm, and the Mahalanobis Distance are special cases of the notion of datum-to-cluster distance defined here. The corresponding cluster models in these cases denote $\psi \in \mathbb{R}^k$, $\psi \in \mathbb{R}^k$, and $\psi$ identifying a pair of a member of $\mathbb{R}^k$ and a $k \times k$ covariance matrix, respectively.

We assume that $\phi(x, \psi)$ is differentiable in terms of $\psi$ and that for any non-empty weighted set $X$, the following function of $\psi$,

$$A_X(\psi) = E\{\phi(x, \psi)\} = \frac{1}{\Omega} \sum_{n=1}^{N} \omega_n \phi(x_n, \psi),$$

(11)

has one and only one minimizer which is also the only solution to the following equation,

$$\sum_{n=1}^{N} \omega_n \frac{\partial}{\partial \psi} \phi(x_n, \psi) = 0.$$  

(12)

In this paper, we assume that a function $\Psi(\cdot)$ is given, which, for the input weighted set $X$, produces the optimal cluster $\psi$ which minimizes (11) and is the only solution to (12). We address $\Psi(\cdot)$ as the cluster fitting function. In fact, $\Psi(\cdot)$ is the solution to the M-estimator given in (11). Examples for $\Psi(\cdot)$ include mean and median when $x$ and $\psi$ are real values and,

$$\phi(x, \psi) = (x - \psi)^2,$$

(13)

and,

$$\phi(x, \psi) = |x - \psi|,$$

(14)

respectively. We note that when a closed-form representation for $\Psi(\cdot)$ is not available, conversion to a W-estimator can produce a procedural solution to (12) (refer to [42,27] for details).

We assume that a function $\Psi(\cdot)$, which may depend on $X$, is given, that produces an appropriate number of initial clusters. We address this function as the cluster initialization function and denote the number of clusters produced by it as $C$. In this work, as discussed in Section 2.4, we make the assumption that the management of the value of $C$ is carried out by an external process which may also intervene between subsequent iterations of the algorithm developed in this paper.

We assume that a robust loss function, $u(\cdot) : [0, \infty] \rightarrow [0, 1]$, is given. Here, the assumption is that,

$$\lim_{\tau \rightarrow \infty} u(\tau) = 1.$$  

(15)

We assume that $u(\cdot)$ is an increasing differentiable function which satisfies $u(0) = 0$ and $u(\lambda) = \frac{1}{\lambda}$, for a known value of $\lambda > 0$, which we address as the scale parameter (note the similarity with the cluster-specific weights in PCM [59]). In fact, $\lambda$ has a similar role to that of scale in robust statistics [44] (also called the resolution parameter [7]) and the idea of distance to noise prototype in NC [18,19]. Scale can also be considered as the controller of the boundary between inliers and outliers [21]. From a geometrical perspective, $\lambda$ controls the radius of spherical clusters and the thickness of planar and shell clusters [61]. One may investigate the possibility of generalizing this unique scale factor into cluster-specific scale factors, i.e. $\lambda_c$ values, in line with $\eta_c$ variables in PCM [59].

In this work, we utilize the rational robust loss function given below,

$$u(x) = \frac{x}{\lambda + x},$$

(16)

Hence, we model the loss of the datum $x_n$ when it belongs to cluster $c$ as,

$$u_{nc} = u[\phi(x_n, \psi_c)].$$

(17)

Note that varying $\lambda$ does not impact the overall geometrical layout of $u(x)$. In fact, $\lambda$ linearly stretches the vertical span of the function. To exhibit this point, we make $\lambda$ explicit in the notation of $u(x)$ and write $u_\lambda(x) = u(\lambda x) = u_1(x)$. We acknowledge that one may consider the possibility of utilizing Tukey’s biweight [6], Hampel [36], and Andrews loss functions in the present framework. Huber and Cauchy loss functions are not bounded and therefore are not applicable to this work. Refer to [57, Table I] for mathematical formulations and to [66] for an implementation which utilizes Huber within the context of FCM.
3.2. Clustering model

As discussed in Section 3.1, a typical clustering algorithm in the literature employs a notion of datum-to-cluster distance which relates an arbitrary datum to an arbitrary cluster. The next step is then to generalize this relationship between an arbitrary datum and a set of clusters. In this framework, the dominant model utilized in the literature is that the clusters "compete" for a datum simultaneously, thus resulting in a weighted-sum-of-distances aggregation model. In effect, this model is utilized in FCM and PCM and the whole spectrum of the approaches which combine and modify these two algorithms in different forms.

As shown in Fig. 1, the classical model is based on a set of clusters which each issues a verdict for any datum and a subsequent consolidation stage which ensures that a sum-of-one constraint, generally in the form shown in (2), is satisfied. A key challenge with this model, however, is that it assumes that every datum is an inlier. While, as discussed in Section 2.3, further developments utilize more complex models in order to address this aspect, these approaches often depend on entities, including model variables and regularization weights, which are hard to perceptually define and challenging to assign a value to. Moreover, these extensions to the classical models are often based on engineered terms and heuristics-driven processes which are later justified through experimental results.

We argue that a core challenge in the classical models is that they utilize clusters as independent points of datum assessment. In other words, in these models, each cluster provides a level of interest in a datum without respect for the relationship between the other clusters and the same datum. In effect, the consolidation stage is required in those models in order to avoid the challenges present in the original PCM model, as discussed in Section 2.3. To our understanding, more efficient execution of cluster-to-cluster "communication" is an avenue which can be explored.

Therefore, we propose the cluster model exhibited in Fig. 2. In this model, a cluster "observes" a datum and makes a decision as to the probability that the datum belongs to the cluster. Note that, in this model, the process of decision-making, which in the classical model is postponed until after the consolidation stage, is also carried out directly by the cluster. We show in this paper that this cluster model allows for the serialization of the clustering process and that it has important benefits at an affordable cost.

3.3. Assessment of loss

We first discuss the membership probabilities corresponding to the clusters in the model depicted in Fig. 9 for the arbitrary datum $x_n$. In order to formalize these probabilities, we define the $C$ sets $X_c$, $c = 1 \ldots C$, where $X_c$ is the set of datums which belong to cluster $c$ when a Maximum Likelihood selection process is carried out. We define the set $X - X$ as the set of outliers. Here,

$$X = \bigcup_{c=1}^{C} X_c.$$  \hspace{1cm} (18)

As seen in Fig. 9, the first cluster decides to "choose" $x_n$ independent of the decisions made by the other clusters. Hence,

$$p\{x_n \in X_1\} = f_{n1}. \hspace{1cm} (19)$$

However, when $c > 1$, we need to include the complete set of probabilities that lead to $x_n$ being passed on to cluster $c$ in the derivations. Hence, for $c = 2$ we write,

$$p\{x_n \in X_2\} = f_{n2}(1 - f_{n1}), \hspace{1cm} (20)$$

and for $c = 3$ we have,

$$p\{x_n \in X_3\} = f_{n3}(1 - f_{n2})(1 - f_{n1}). \hspace{1cm} (21)$$

Fig. 1. The classical clustering model utilized in FCM, PCM, and a significant number of other approaches in the literature. Here, the distances between a datum and a set of clusters are examined in parallel and a consolidation stage ensures that a sum-of-one constraint is satisfied.
Similarly,
\[ p\left\{ x_n \in \mathbf{X} \right\} = f_n \prod_{c=1}^{C-1} (1 - f_{nc}). \] (22)

Therefore, the probability that \( x_n \) is an outlier is calculated as,
\[ p\left\{ x_n \in \mathbf{X} \right\} = 1 - \sum_{c=1}^{C} p\left\{ x_n \in \mathbf{X}_c \right\} = 1 - \sum_{c=1}^{C} f_{nc} \prod_{c=1}^{C-1} (1 - f_{nc}) = \frac{C}{C-1} (1 - f_{nc}). \] (23)

There are multiple methods for proving the last equality in (23). One may do the derivation through defining,
\[ F_{c_1c_2} = 1 - \sum_{c=c_1}^{c_2} f_{nc} \prod_{c_1}^{C-1} (1 - f_{nc}) \],
and subsequently utilizing reverse induction in order to start from \( F_{Cc} = 1 - f_{nc} \) and to calculate \( F_{1c} \).

We use these results in order to assess the aggregate loss in the clustering model shown in Fig. 9 corresponding to the arbitrary datum \( x_n \). We first write,
\[ E\{\text{Loss}|x_n\} = \sum_{c=1}^{C} E\{\text{Loss}|x_n \in \mathbf{X}_c\}p\left\{x_n \in \mathbf{X}_c\right\} + E\{\text{Loss}|x_n \in \mathbf{X} - \mathbf{X}\}p\left\{x_n \in \mathbf{X} - \mathbf{X}\right\}. \] (25)

Then, we use (22) and (23) and rewrite (25) as,
\[ E\{\text{Loss}|x_n\} = \sum_{c=1}^{C} \left\{ \sum_{n=1}^{N} \omega_n \left[ f_{nc} \prod_{c=1}^{C-1} (1 - f_{nc}) \right] \right\} + \frac{1}{\Omega} \sum_{n=1}^{N} \omega_n \prod_{c=1}^{C} (1 - f_{nc}). \] (26)

Here, the loss of an outlier is modeled as 1, the value to which the robust loss function converges to when \( \phi(x_n, \psi_c) \) is very large.

Now, we use (10) and combine the single-datum loss model derived in (26) for all datums in order to estimate the aggregate loss in the system as,
\[ E\{\text{Loss}|\mathbf{X}\} = \frac{1}{\Omega} \sum_{n=1}^{N} \omega_n \left[ f_{nc} \prod_{c=1}^{C-1} (1 - f_{nc}) \right] + \frac{1}{\Omega} \sum_{n=1}^{N} \omega_n \prod_{c=1}^{C} (1 - f_{nc}). \] (27)

Hence, we construct the clustering algorithm developed in this paper as an optimization problem in which the following objective function is to be minimized,
\[ A = \sum_{n=1}^{N} \sum_{c=1}^{C} \omega_n \left[ f_{nc} \prod_{c=1}^{C-1} (1 - f_{nc}) \right] u_{nc} + C^{1-m} \sum_{n=1}^{N} \sum_{c=1}^{C} \omega_n \prod_{c=1}^{C} (1 - f_{nc}). \] (28)

Here, we have dropped the constant \( \Omega^{-1} \) factor and we have chosen to rewrite the constant factor 1 as \( CC^{-1} \) for reasons which will become clear later.

Note that, in (28), the only requirement for the value of \( f_{nc} \) is that \( f_{nc} \in [0, 1] \). In other words, a sum-of-one constraint, similar to the one given in (2), is not required here. The relaxation of this requirement in the method developed in this paper, however, is distinctly different from the approach in PCM, as in this work the clusters are inherently tied together. We emphasize that not only the disadvantageous separation of clusters, as observed in the original PCM, is not present in this work, but also that a concept of outlier rejection emerges within the model developed in this paper. Additionally, as will be shown later, the condition \( f_{nc} \in [0, 1] \) will be organically satisfied in this work.

Close assessment of (28) shows that this cost function complies with an HCM-style hard template. It is known, however, that migration from HCM towards a fuzzy/probabilistic formulation has important benefits, as outlined in Section 2.1. Hence, defining \( U = C^{1-m} \), we rewrite the aggregate loss as,
\[ A = \sum_{n=1}^{N} \sum_{c=1}^{C} \omega_n \left[ f_{nc} \prod_{c=1}^{C-1} (1 - f_{nc}) \right] u_{nc} + U \sum_{n=1}^{N} \omega_n \prod_{c=1}^{C} (1 - f_{nc}). \] (29)

![Fig. 2. Single cluster model utilized in the algorithm developed in this paper.](image-url)
3.4. Alternating Optimization strategy

In this section, we provide a solution strategy for minimizing (29). This process utilizes a set of input clusters and carries a variant of the well-known Alternating Optimization (AO) methodology. This process is composed of a number of subsequent stages. In each step, the objective function is minimized relative to a subset of the clustering unknowns. In fact, the procedure developed in this work contains two stages of assignment and reclustering. During the assignment stage, the cluster representations are assumed to be known and the optimal values of \( f_{nc} \) are estimated. The reclustering stage, on the other hand, fits optimal clusters to the set of datums given their known membership status to the clusters. Here, these two processes are described.

The classical approach to utilizing AO in the context of fuzzy clustering is to work on the partial derivative of \( \mathcal{A} \) relative to \( f_{nc} \) while (2) is incorporated into the objective function using Lagrange Multipliers. However, because the structure of (29) is different from the classical objective functions present in the fuzzy/possibilistic literature, a different approach is required in the present work. Additionally, (2) is not applicable to the present work.

We define,

\[
A_{c*}(n) = \sum_{c \in \mathcal{C}} \left[ f_{nc}^m \left( 1 - f_{nc} \right)^m \right] u_{nc} + \sum_{c \in \mathcal{C}} \left( 1 - f_{nc} \right)^m. \tag{30}
\]

One may in fact show that \( A_{c+1}(n) = U \) and, using (29), that,

\[
A = \sum_{n=1}^{N} \omega_n A_1(n). \tag{31}
\]

Now, direct derivation shows that,

\[
A_{c*} = f_{nc}^m + (1 - f_{nc})^m + 1(n). \tag{32}
\]

Note that an important result of (32) is that \( A_{c+1} \) is independent of \( f_{nc} \).

Now, we address the process of minimizing \( A(n) \), i.e. contribution of \( x_n \) to \( \mathcal{A} \), through finding the optimal values for \( f_{nc} \) for \( c = 1, \ldots, C \). We have,

\[
\frac{\partial}{\partial f_{nc}} A(n) = \frac{\partial}{\partial f_{nc}} A_1(n) = \frac{\partial}{\partial f_{nc}} (f_{n1} u_{n1} + (1 - f_{n1})^m A_2(n)) = (1 - f_{n1})^m \frac{\partial}{\partial f_{nc}} A_2(n). \tag{33}
\]

Continuing this derivation proves that,

\[
\frac{\partial}{\partial f_{nc}} A(n) = \prod_{c=1}^{c=1} \left( 1 - f_{nc} \right)^m \frac{\partial}{\partial f_{nc}} A_{c*}(n) = \prod_{c=1}^{c=1} \left( 1 - f_{nc} \right)^m \left[ mf_{nc}^m u_{nc}^m - m(1 - f_{nc})^{m-1} A_{c+1}(n) \right]. \tag{34}
\]

Equating (34) with zero yields,

\[
f_{nc} = \left( \frac{A_{c+1}(n)^m}{u_{nc}^m + A_{c+1}(n)^m} \right) \tag{35}
\]

Here, and throughout the derivations, we use the following notation in order to simplify the formulations,

\[
m = \frac{1}{m - 1}. \tag{36}
\]

Now, we plug the value of \( f_{nc} \) from (35) in (32) and produce the following simplified update equation,

\[
A_{c*}(n) = (1 - f_{nc})^{m-1} A_{c+1}(n). \tag{37}
\]

Hence, we propose the procedure shown in Algorithm 1 for calculating \( f_{nc} \).

**Algorithm 1.** The procedure for calculating \( f_{nc} \).

```plaintext
for n = 1 to N do
    A = C \cdot m;
    for c = C down to 1 do
        f_{nc} = \frac{A_{c+1}(n)^m}{u_{nc}^m + A_{c+1}(n)^m};
    A = (1 - f_{nc})^{m-1} A;
end
```
Note that, as discussed in Section 2.1, there is general consensus in the community that $m = 2$ is an appropriate choice for the value of the fuzzifier. Setting $m = 2$ results in $m - 1 = \tilde{m} = 1$, which simplifies many of the formulations given in this paper. Due to shortage of space we do not carry these simplified equations. Note that one may utilize the Klauwnn–Hoppner generalization [53] of the fuzzifier within the context of the algorithm developed in this paper. Also, one can directly show that this process always guarantees that $0 \leq f_{nc} \leq 1$, as required by the model.

Now, we develop the update procedure for the clusters. This stage is similar to the classical FCM. We write,

$$\frac{\partial}{\partial \psi_c} \partial \mathbf{A} = \sum_{n=1}^{N} \omega_n \left[ \sum_{c=1}^{C} \left( 1 - f_{nc} \right)^m \right] \frac{\partial}{\partial \phi} \phi(x_n, \psi_c).$$  \hspace{1cm} (38)

Here, we have used the substitute notation,

$$u_{nc} = \frac{d}{d\psi} u_{|\psi(x_n, \psi_c)|},$$  \hspace{1cm} (39)

Now, using (12) we have,

$$\psi_c = \mathcal{A} \left( \left\{ \omega_n f_{nc}^m u_{nc} \prod_{c=1}^{C} \left( 1 - f_{nc} \right)^m, x_n \right\} \right).$$  \hspace{1cm} (40)

Note, however, that the closed form given in (40) is based on ignoring the dependence of $u_{nc}$ on $\psi_c$. Hence, we propose to utilize (40) in order to produce the new cluster representation $\psi_c^n$. Then, we only use this cluster representation if replacing the current $\psi_c$ with $\psi_c^n$ results in a reduction in the value of $\mathbf{A}$.

An alternative method is to utilize the technique developed by Weiszfeld [95] and Miehle [70,84] (similar techniques are cited under different names as well [62,16]). The Weiszfeld technique utilizes the fixed point method in order to solve (12) when $\phi(\cdot)$ is not the Euclidean distance function (refer to [62] for details and to [22] for options for the acceleration of the technique). A weighted version of the Levenberg–Marquardt algorithm [71] may also be applicable for certain distance functions and loss functions.

3.5. Outlier rejection

An interesting observation about the model developed in Section 3.2 is the emergence of the notion of the probability that a datum is an outlier. In effect, the model developed in this paper defines $f_{nc}$ variables as conditional probabilities of belonging to the clusters. However, through the process, the probability of belonging to none of the clusters emerges, which we derive an analytic form for in (23).

We propose to utilize a Maximum Likelihood inference framework and to compare the probability values corresponding to $x_n \in \mathbf{X}$, for $c = 1, \ldots, C$ as well as the probability that $x_n \in \mathbf{X} - \mathbf{X}$. Derivation shows that a datum belongs to one of the $\mathbf{X}_c$, and therefore is an inlier, when,

$$\max_c \left\{ f_{nc} \prod_{c=1}^{C} \left( 1 - f_{nc} \right) \right\} > \max_{c} \left( 1 - f_{nc} \right).$$  \hspace{1cm} (41)

Here, we have used (22) and (23).

Hence, if the problem is to be solved in inclusive mode, then datum $x_n$ is assigned to cluster $c_n$ where,

$$c_n = \arg \max_c \left\{ f_{nc} \prod_{c=1}^{C} \left( 1 - f_{nc} \right) \right\},$$  \hspace{1cm} (42)

In non-inclusive mode, however, $c_n$ is not defined if $x_n$ is not an inlier, i.e. if it does not satisfy (41). This strategy has similarities to Conditional Fuzzy C-means (CFCM) [64], but the method developed in this paper does not require the a priori knowledge necessary in CFCM.

4. Experimental results

A common practice in many clustering works is to provide experimental results as evidence for the superiority of the methods developed therein. We argue that this after-the-fact evaluation approach is a direct implication of the fact that many of these works utilize objective functions and/or constraints which are selected based on intuition. In other words, our argument is that the methodology utilized when the clustering problem is formalized must be validated as well. Hence, whether a particular approach is capable of producing acceptable results for a certain number of problems is a weak indicator of its usability. We find the necessity of validating the construction methodology behind a clustering problem to be important from an epistemological perspective.
As presented in Section 3, this work constructs the formalization of the clustering problem through Bayesian inference. In this process, we actively avoid the use of our intuition in generating variables, the objective function, and constraints, and strictly require the formalization to comply with an explicitly stated mathematical model. This is in direct contrast to FCM, PCM, and a majority of other methods in the field, which despite their magnificent achievements, are in essence formalizations based on what the respective designers of the algorithms have found to be appropriate.

Nevertheless, in this section, we provide sample experimental results produced through the application of the method developed in this paper on multiple problem instances within the context of three problem classes. In each case, we provide comparisons with the outcomes of the classical FCM.

Fig. 3. Results of the execution of the developed method for grayscale image multi-level thresholding. (a) Developed algorithm. (b) Classical FCM.
Note that the two algorithms are always executed using identical input datums and there is no difference in parameters or strategies utilized by the two algorithms. In effect, the comparison provided in this section is at the level of objective functions. Nevertheless, we emphasize that this presentation is by no means intended to be a complete evaluation process. As such, the contribution of this paper is a novel modeling framework and the emergent properties of that. Hence, the carried results are to be treated as samples utilized in order to exhibit the differences of the previous model and the presented one.

First, in Section 4.1, a few notes on the implementation of the developed algorithm are given. Then, in Sections 4.2–4.4, the deployment of the developed algorithm for three different problem classes are discussed.

Fig. 4. Results of the execution of the developed method for grayscale image multi-level thresholding. (a) Developed algorithm. (b) Classical FCM.
4.1. Implementation notes

The developed algorithm is implemented as a class named Belinda in MATLAB Version 8.1 (R2013a). It takes use of Image Processing Toolbox for minor image-related primitives and the major operations are implemented as C/MEX dll’s. The code is executed on a Personal Computer which runs Windows 7, 64 bit, on an Intel Core i5-2400 CPU, 3.10 GHz, with 8.00 GB of RAM.

Each problem class is implemented as a child class for Belinda. The child classes implement a constructor which creates the weighted set $X$ based on the input image, data file, etc. The child classes also implement the three functions $\phi(\cdot)$, $\Psi(\cdot)$, and $\Psi(\cdot)$ and set the value of $\lambda$ independent of $X$. The child classes are not responsible for any of the core operations of the developed algorithm. These operations are implemented in the parent class Belinda. This class abstracts datums and clusters and operates on them indifferent to the content of the problem class-specific entities.

4.2. Grayscale image multi-level thresholding

The problem of grayscale image multi-level thresholding defines datums as grayscale values and models a cluster as an interval on the grayscale axis centered at the scalar $w_c$ [48].

In order to produce the datums, we calculate the histogram of the input image (32 bins). Here, each bin represents one datum and the number of pixels in the bin, normalized over the area of the image, produces the corresponding weight. Distance between a datum and a cluster is defined as the square difference and the initial clusters are defined as

![Fig. 5. Results of the execution of the developed method for 2-D Euclidean clustering. Size of the datums denote their weight and their shade of gray indicates the probability of being an inlier. (a) Developed algorithm. (b) Classical FCM.](image-url)
uniformly-distributed points in the working range. The cluster fitting function in this case calculates the weighted sum of the input elements. This problem class utilizes a scale of 25 gray levels and is defined as inclusive.

Fig. 3 presents the results of running the developed algorithm on the standard image Cameraman alongside the results produced by the classical FCM for the same sample image. Here, in visualization, \( x_n \) is replaced with \( \psi_n \) in order to exhibit the clusters. We note that meaningful comparison of the two outputs in the image domain is challenging. Therefore, we review the layout of the clusters in the histogram domain.

The two graphs given in Fig. 3 visualize the histogram of the input image overlaid by the converged clusters as well as the probabilities that the datums are inliers. In these visualizations, the thick dashed line denotes the histogram of the input image, thus effectively visualizing the values of \( \omega_n \). The thin dashed line exhibits the possibility of being an inlier for \( x_n \), which, as expected, is always one for FCM. The colored curves in these figures denote membership to the converged clusters.

We first note that FCM clusters are bulkier than those produced by the proposed algorithm, as demonstrated by wider range of high membership values in Fig. 3(b)-bottom compared to Fig. 3(a)-bottom. Therefore, we argue that the proposed method produces clusters which are more focused on a homogenous subset of the input datums. A side achievement of this phenomenon is that FCM clusters claim high membership values to outliers. This is for example visible in the right end of Fig. 3(b)-bottom, where the curve representing the third cluster resides on a significant height. We compare this situation to the low level of membership claimed within the context of the developed algorithm to the same set of datums (see the right end of Fig. 3(a)-bottom).

This result also indicates the difference in the computational complexity of the developed method compared to the classical FCM. In fact, the developed method takes more iterations to converge (27 for the developed algorithm vs. 11 for classical FCM) and also takes more time to settle into the final solution (100 ms for the developed algorithm vs. 17 ms for classical FCM).

![Fig. 6. Results of the execution of the developed method for 2-D Euclidean clustering. Size of the datums denote their weight and their shade of gray indicates the probability of being an inlier. (a) Developed algorithm. (b) Classical FCM.](image-url)
FCM). This is due to the more complicated nature of the calculation of $f_{nc}$ in the developed algorithm as well as the appropriateness check required after each call to $\Psi(\cdot)$ in this algorithm. In fact, in an experiment which involved the execution of the two algorithms on 212 standard images, adopted from [2] and USC-SIPI, the developed algorithm required $98 \pm 57$ ms to complete while the classical FCM converged in $35 \pm 16$ ms. Thus, the developed algorithm is in this case about 3.15 times more expensive than the classical FCM. The input images in this experiment contain $512 \times 512$ pixels.

Fig. 4 shows the output generated by the developed algorithm compared to that of the classical FCM for the standard image Lena. This comparison validates the same observations made in Fig. 3, i.e. the clusters generated by the developed algorithm are more specific and that FCM clusters claim high membership to outliers.

4.3. 2-D Euclidean clustering

The problem of 2-D Euclidean Clustering is the classical case of finding dense clusters in weighted 2-D data. Here, dense is defined in terms of inter-cluster proximity of points.

In this problem class, the two functions $\phi(\cdot)$ and $\Psi(\cdot)$ are the Euclidean and the center of mass operators and the set of datums is generated based on the superposition of multiple Gaussian distributions. This problem class is defined as non-inclusive and utilizes the scale of 4.

Fig. 5 shows the results of executing the developed algorithm as well as those of the classical FCM on a set of datums. Here, the colored polygons exhibit the convex hull of the datums in each cluster and shades of gray indicate the probability that the corresponding datum is an inlier. Radius of a datum conveys its weight.

![Fig. 7. Results of the execution of the developed method for plane finding in range data. (a) Developed algorithm. (b) Classical FCM.](image-url)
We note that while the developed algorithm produces compact clusters, the classical FCM extends the clusters into the entire set of datums. This is a direct result of the outlier assessment process organically embedded in the developed algorithm. Nevertheless, we emphasize, that the outlier rejection mechanism present in the developed algorithm is an emergent property of the utilized clustering model. This, we argue, is a major advantage over the approaches in the literature in which a mechanism involving confidence-thresholding is utilized in order to detect outliers. Such methods require the "prudent" [31, p. 452, top of column two] setting of a configuration parameter or the approach will fail to perform outlier rejection efficiently. This phenomenon is also visible in the different shades of gray given to the different datums in the results of the developed algorithm. As seen in Fig. 5(a), darker points are considered more probable to be inliers by the developed algorithm. Hence, we observe a more efficient detection of the boundaries of the clusters and a more optimal separation of the datums into inliers and outliers by the developed algorithm, compared to the classical FCM.

The developed algorithm converges in this case after 19 iterations and 189 ms of operations while the classical FCM requires 31 iterations and 190 ms. These values are not representative however, because experimentation with 100 different sets of datums indicates that the developed algorithm requires 215 ± 81 ms of operation while the classical FCM concludes in 124 ± 56 ms. Hence, in average, the developed algorithm inflates the execution time by a factor of 2.03, compared to the classical FCM. Fig. 6 shows another set of datums processed by the developed algorithm as well as by FCM.

4.4. Plane finding in range data

In this section, we discuss the problem class which is concerned with finding planar sections in range data. The input set of datums in this problem class contains 3-D points captured by a Kinect 2 depth acquisition device. The depth-maps used in this experiment are captured at the resolution of 512 × 424 pixels. Note that, the intrinsic parameters of the camera are acquired through the Kinect SDK and that each datum in this problem class has the weight of one.

Clusters in this problem class are defined as planes which have a thickness and the process of fitting a plane to a weighted set of 3-D points is carried out through a weighted variant of Singular Value Decomposition (SVD). This problem class utilizes

![Fig. 8. Results of the execution of the developed method for plane finding in range data. (a) Developed algorithm. (b) Classical FCM.](image-url)
the scale of 200 mm and is defined as non-inclusive, because we expect outliers and other datums which do not belong to any of the planes to exist in the set of datums and wish to identify and exclude them. We initialize the process by placing three planes in the extremities of the room, where based on a priori knowledge we expect planes to exist.

Fig. 7 shows the results of executing the developed algorithm as well as the classical FCM on range data captured from a room with visible floor and walls and three human figures present in it. Here, the pointcloud is viewed from the top while the camera is located at the bottom center of the view.

In Fig. 7(a) and (b), points are painted according to the cluster they belong to. We observe in Fig. 7(a) that the floor and the two walls are successfully detected by the developed algorithm. Additionally, points belonging to the people present in the scene are discarded by the developed algorithm as outliers, and are hence painted as black. Close assessment of Fig. 7(b), however, reveals that the human figures are considered as parts of the different planes detected by the classical FCM. The fact is that FCM has completely failed in detecting the planes present in the scene and has converged to arbitrary locations in the data, except for the floor plane which is partly detected. Nevertheless, this plane is also mixed with portions of one of the walls as well. On the contrary, Fig. 7(a) indicates that the developed algorithm robustly detects and locks into the two walls in the scene and the floor. A similar observation is made in other sets of datums, which belong to the same scene captured from different vantage points, as shown in Fig. 8. As seen here, in every case, the developed method correctly identifies the planes present in the scene while the classical FCM always fails.

For the results shown in Fig. 7, the developed algorithm converges after 52 iterations and 4228 ms of operations while classical FCM requires 60 iterations and 4329 ms before it concludes. We subsequently utilized a set of 7 samples taken at the same room, where the camera was placed in different positions and the human subjects moved around. In average, the developed algorithm required 2900 ± 819 ms while the classical FCM converged after 2171 ± 1372 ms. Hence, in average, the computational complexity of the developed algorithm is 1.72 times more than that of the classical FCM. This numerical assessment should be reviewed while the failure of FCM in the experiments is also taken into consideration.

5. Conclusions

In the past few decades, many approaches to solving the fuzzy/possibilistic clustering problem have been proposed. These different methods utilize different sets of identifiers in order to model concepts such as membership to a cluster, probability of being an inlier, and others. We argue, however, that the literature is dominated by approaches which utilize parallel clustering models. In essence, in these models, a group of clusters process a datum independently, and the results are then passed through a consolidation stage which satisfies constraints such as sum-of-one for membership values. We also argue that works in the literature are generally heavily based on heuristically engineered objective functions and constraints which are intuitively translated from a verbal description of what the clustering algorithm is expected to carry out.

In this work, we develop a novel sequential clustering model. In this model, clusters receive the datums in a sequence and either accept it or pass it on to the next stage. We outline a Bayesian inference framework which we use in order to assess the loss in the model. The assessment carried out in this paper is independent of the datum and cluster models of any particular problem class.

We observe that the notion of the probability of being an inlier emerges within the developed clustering model. Hence, in this work, a generic class-independent clustering algorithm is developed which also carries out outlier rejection, if required, with no dependence on any confidence threshold or similar measures.

We observe that, conventionally, suggestions of novel clustering approaches attempt to attain validation through providing experimental results which indicate that the developed technique in fact yields acceptable results. We argue that this after-the-fact approach, while necessary, is based on an epistemological challenge. In this argument, we suggest that the translation of verbal expectations from the clustering problem into formal notations must follow a clear mathematical model. As such, while providing experimental results in this context is a necessity, we argue that such results, alone, do not validate a method in the absence of a sound derivation procedure.

Nevertheless, we considered three problem classes and outlined the utilization of the developed method for them. In fact, for each problem class, we presented a number of problem instances and compared the results of the developed algorithm on these samples to what is produced by the classical FCM. In this work, the developed algorithm and the classical FCM were executed in completely identical circumstances.
We observed that the classical FCM extends the clusters in order to cover the entire set of datums, whereas the developed method utilizes its emerged outlier rejection mechanism in order to detect and exclude outliers. We also showed that the robust loss function utilized in the developed algorithm allows it to focus on dense clusters and to avoid adjacent distractions. These benefits are achieved by the developed algorithm while its execution time is less than four times that of the classical FCM. In fact, in certain cases, the developed algorithm functions faster than the classical FCM, mainly because convergence is achieved in a fewer number of iterations by the developed algorithm compared to the classical FCM.

Acknowledgments

The author wishes to thank the management of Epson Edge for their help and support during the course of this research. We wish to thank Mahsa Pezeshki for proofreading this manuscript. The idea for this research was conceived at The 3 Brewers pub in Toronto while enjoying the night with a group of friends. We wish to thank the respective anonymous reviewers of this paper for their constructive comments and recommendations. We wish to thank friends on Twitter, Facebook, and Google+ for their help in finding key references for this paper.

References