Fuzzy Clustering with M-Estimators

K Madhava Krishna[1] Prem K Kalra[2]

1 International Institute of Information Technology
Hyderabad – 500 019, INDIA

2 Department of Electrical Engineering
Indian Institute of Technology, Kanpur, INDIA

mkrishna@iiit.ac.in kalra@iitk.ac.in

Abstract. We present an extension of the FCM over the loss functions used in the M-estimators of robust statistics akin to the generalization of the fuzzy-C-means algorithms over the norm distances [1]. The effect of these estimators in reducing the bias of the outliers while estimating the cluster prototypes are studied and compared. The comparisons have been done over synthetic data as well as simulated data consisting of range sensor readings representative of the objects in the neighborhood of a navigating mobile robot. For the synthetic data set the comparison is attempted over the popular FCM algorithm. For the sensory data set the Adaptive Fuzzy Clustering (AFC) algorithm [2] has been employed and extended over the loss functions of robust statistics. The AFC is utilized considering the shape of the objects encountered by the robot in a typical indoor environment.

1 Introduction

Real world data is noisy and interspersed with outliers. The effects of outliers on estimates of parameters are well known and analyzed over centuries [3, 4]. Earlier philosophy seems disposed towards a rejection paradigm that favors throwing away values that are far away from the main group of data even if these values are actually good data providing information. It was felt that the loss in accuracy of the experiment by throwing away a couple of good values is small compared to the loss incurred by keeping even one bad value [5]. Nonetheless the presence of outliers can be a source of extra information that one is not inclined to discard. With the developments in robust methods of estimation one need not resort to either of the two risks. Robust inference techniques can employ all the available data while minimizing the effect of outliers simultaneously when computing prototype estimates from the distribution.

The sensitivity of the standard squared Euclidean norm to outliers when used in objective functions for prototype estimation is well known [6]. Hence more robust

Bhanu Prasad (Editor): IICAI-05, pp. 2101-2119, 2005.

distance measures are designed to improve resistance to outliers. In regression analysis the weighted medians [6] provides a robust slope estimate when compared with the least mean of squares method [7]. The weighted median has found its applications in risk management [8] and image processing [9]. The famous and ubiquitous FCM algorithm of Bezdeck [10] is also sensitive to outliers due to its employing the squared Euclidean norm in its objective functional. While the FCM has proven itself useful in numerous applications [11] the quality of the computed cluster centers can be degraded due to the presence of outliers in the data set. This is because the weighted squared Euclidean metric can assign significant memberships to outlying data points pulling the cluster prototypes from the center or the main distribution of the cluster. The Fuzzy C Median (FCMED) algorithm [12,13] had been proposed as a robust alternative to the FCM algorithm, which has been recently extended over generalized L_p norm distances by Hathaway and Bezdeck [1].

The modifications to the fuzzy clustering algorithms over the robust loss functions of M-Estimator theory has been analyzed to a considerable extent by Krishnapuram. For example the Robust-C-prototype algorithm [14] makes use of the M-Estimator in [15], the fuzzy trimmed C prototype algorithm [16] makes use of the least trimmed square estimator [17] while the fuzzy-C-least median of squares algorithm uses the least median of square estimators. A holistic view of robust clustering methods had also been given in [18].

In this paper the fuzzy clustering algorithm has been further extended over the M-Estimators that make use of a modified form of “Fair”, $L_1 - L_2$, Cauchy and Huber’s loss functions. The effect of incorporating the un-squared Euclidean norm in place of the standard squared Euclidean norm has also been investigated. A comparative study has been carried across these estimators vis-à-vis the previously employed L_p norm measures when they are incorporated into the FCM algorithm tested on synthetic data embedded with outliers. As a possible engineering application the efficacy of the estimators are investigated over data obtained from range sensors of a mobile robot navigating in a simulated environment. The sensor readings of a robot are prone to noise and outlier ridden and thus lends itself as a test bed for performance validation of the robust clustering algorithms. The robust clustering algorithms can have a prominent role to play in estimating the prototypes of the objects in the robot’s neighborhood especially since data is expected to be outlier ridden. Here the robust loss functions of the M-Estimators are incorporated into the objective functional of the AFC [2] algorithm. The AFC has been employed here considering the typical shape of the objects encountered in an indoor environment.

The rest of the paper has been organized as follows. In section 2 a brief overview of M-Estimator theory and the extension of the fuzzy clustering algorithms over the robust loss functions is presented. Section 3 presents the simulation analysis over synthetic data while section 4 deals with an application in robot navigation. Section 5 concludes with the discussions.

2 Integrating Fuzzy Clustering with M-Estimators

2.1 A Brief Overview of M-Estimators:

The M-Estimators are generalizations of the maximum likelihood estimators and are solutions to the general structure:

$$\min\Big(\sum_{k=1}^{N} \rho(r_k) \Big) \qquad \textbf{(1)}$$

Here r_k can be a residual of the form $r_k = x_k - v$ between the feature vector x_k and the prototype estimate v or $r_k = dist(x_k - v)$, where $dist(.)$ is any distance measure between x_k and v. The function ρ is a robustifying function that helps in reducing the effect of outliers in the standard formulation of least squares method that tries to minimize

$$\sum_k r_k^2 \qquad \textbf{(2)}$$

by replacing (2) with the form of (1). ρ is known as the loss function. Conditions of robustness require that the parameter v to be minimized have a unique minimum [19]. Equivalently this entails that the loss function ρ be convex in v or the second derivative of ρ is non-negative definite.

The problem of equation (1) can be recast into iterated weighted least squares one as follows. Let $v = [v_1, v_2, ..., v_s]^T$ be the parameter vector to be estimated. Then the M-Estimator of v based on the function $\rho(r_i)$ is the vector $\hat{v}$, which is a solution of the following s equations of the form

$$\sum_k \psi(r_k) \frac{\partial r_k}{\partial v_j} = 0 \qquad ; j = 1,2,..., s \qquad \textbf{(3)}$$

The function $\psi(r) = \frac{\partial \rho}{\partial r}$ is called the influence function, which needs to be bounded for a robust estimate [19]. If v_j consists of a single term, then $\frac{\partial r_k}{\partial v_j}$ is unity and can be removed and the resulting M-estimate is called the robust estimate of the location of the sample. If $\rho(r) = r^2$ the resulting estimate is the sample mean and if $\rho(r) = |r|$ the resulting estimate is the sample median. Since ρ is convex the form of equation (1) or a weighted form of (1) such as

$$\sum_k u_k \rho(r_k) \tag{4}$$

for positive weights is also convex and hence possesses a unique minimum. The condition for unique minimum makes (1) or (4) or equivalently (3) amenable to a numerical approximation of the estimate $\hat{v}$ through methods such as bisection or Newton Raphson.

An alternative and less tedious approach that circumvents the need to find out the derivatives of (3) is through the method of W-estimators as follows. Let $w(r)$ be defined as $rw(r) = \psi(r)$. Then (3) can be rewritten as

$$\sum_k w(k) r_k \frac{\partial r_k}{\partial v_j} = 0, \quad j = 1,2,\ldots,s \tag{5}$$

If $r_k = C(x_k - v_j)$ then

$$v_j = \frac{\sum_k w(r_k) x_k}{\sum_k w(r_k)} \tag{6}$$

Thus v_j becomes the weighted mean of x_k that can be solved iteratively. Here $w(r)$ is called as the weight function and the estimate of v_j so obtained is known as the W-estimate of v_j.

2.2 Incorporating M-Estimator into Fuzzy Clustering

The FCM tries to minimize

$$J(U,V) = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^m \|x_k - v_i\|_2^2 \tag{7}$$

with $\sum_{k=1}^{c} u_{ik} = 1$ and $0 \le \sum_{k=1}^{N} u_{ik} \le N$, u_{ik} is a member of the well-known fuzzy partition matrix. The form of equation (7) has a formulation similar to that of the standard least squares in (2). Hence in the same vein as the least squares estimate, the estimate of the prototypes $v_i, i = 1,..., c$ by minimizing (7) is susceptible to large errors when the data set consists of outliers. Modifications to (7) by making the estimates robust include the generalized L_p norm distances [1], which tries to minimize

$$J(U,V) = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m} \left\| x_k - v_i \right\|_p^p, \quad 1 \le p \le 2 \qquad \textbf{(8)}$$

The minimization of (8) proceeds across a $c \times s$ decoupled independent univariate minimizations of the form:

$$f_{ji}(v_{ji}) = \sum_{k=1}^{N} u_{ik}^{m} \left| x_{jk} - v_{ji} \right|^p, \; j = 1,2,.., s; \; i = 1,2,..., c \qquad \textbf{(9)}$$

The form of (9) is convex for $p \in [1, \infty)$. For $p = 1$, it is piecewise linearly convex as shown in figure 1a and for $p > 1$ it is smooth as shown in figure 1b for $p = 1.2$. Since f_{ji} possesses a unique minimum for $p \ge 1$, a solution to the unique zero of the first derivative of (9) can be found. A brief summary of the FCM algorithm is given in the appendix.

As was the case with the least squares formulation, the FCM formulation of (7) can be robustified by introducing the loss function ρ such that

$$J(U,V) = \min \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik} \rho(r_{ik}) \qquad \textbf{(10)}$$

which can as usual be decomposed into c univariate distributions of the form

$$J_i = \min \sum_{k=1}^{N} u_{ik} \rho(r_{ik}), \quad i = 1,2,..., c \qquad \textbf{(11)}$$

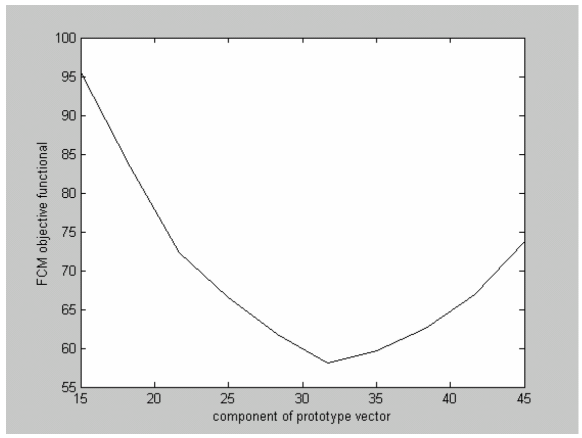


Fig. 1: Plot of FCM objective functional for L_1 norm vs. a component of the prototype vector. The function is piecewise linearly convex.

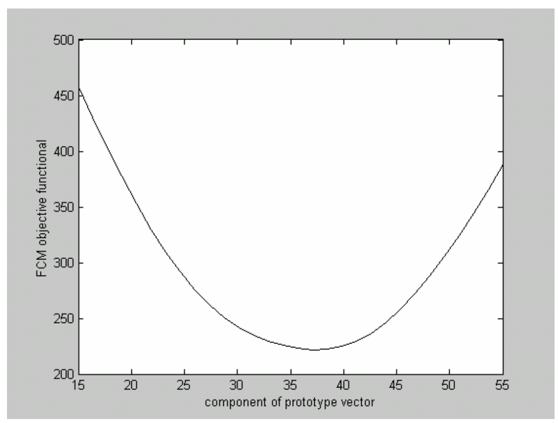


Fig. 2. Plot of objective function for $L_{1.2}$ norm. The function is everywhere differentiable and convex.

Here r_{ik} can be some distance function d_{ik} such as $d_{ik} = \|x_k - v_i\|_2^2$. The primary drawback of the L_2 norm in the objective functional of (7) is its tendency to get influenced considerably by datum to prototype dissimilarities. It is also to be noted that the influence function of the L_2 loss function is linear and unbounded. The influence function measures the influence of the datum on the prototype to be estimated. The influence of the datum on the prototype increases linearly as the residual value increases there by pulling the estimates towards the outliers significantly confirming the non-robustness of the L_2 estimator. It can be shown that the influence functions introduced below are bounded, thereby the growth of the residual error with the growth in datum to prototype dissimilarity is curtailed increasing the robustness of the estimate. The other M-Estimators incorporated into the FCM functional are as follows:

2.2.1 L_1 - L_2 estimator [20]

The loss function is defined as

$$\rho(r) = 2\left(\sqrt{1+r^2/2} - 1\right) \tag{12}$$

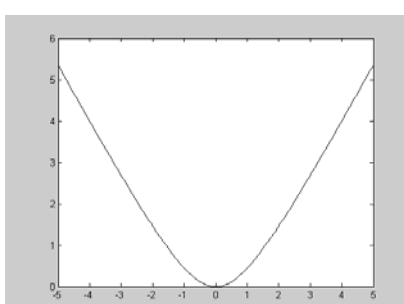
where $r = x - v$ for the unidimensional case with the influence function as $\frac{r}{\sqrt{1+r^2/2}}$ and the weight function as $1\big/\sqrt{1+r^2/2}$. The graphical form of the loss function in figure 2a immediately reveals its convexity for the unidimensional case. An extension to multi-dimension takes the form $\rho(d_{ik}^2) = 2\left(\sqrt{1+d_{ik}^2/2} - 1\right)$ where

$$d_{ik}^2 = \left\| x_k - v_i \right\|_2^2 = \sum_{j=1}^{s} (x_{kj} - v_{ij})^2 \tag{13}$$

When (13) is incorporated into the FCM objective functional it takes the form

$$J_i = \sum_{k=1}^{N} u_{ik}^m \rho(d_{ik}^2), i = 1,..., c \tag{14}$$

The graphical form of (14) in figure 2b is once again convex in the parametric space thus entailing an unique solution in that space. First derivatives of (14) in the components of v_i gives rise to coupled non linear equations in the components that can be approximated through the Newton Raphson's method. The $L_1 - L_2$ estimators take advantage of L_1 estimate to reduce the influence of large errors and the convex



nature of L_2 estimates.

Fig. 2a: Plot of $L_1 - L_2$ loss function for single dimensional case

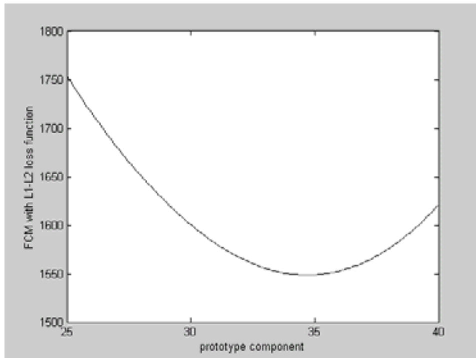


Fig. 2b: Plot of FCM objective functional modified with $L_1 - L_2$ loss function.

2.2.2 "Fair" Estimator [21,22]

For the unidimensional case the loss function is of the form

$$\rho(r) = C^2\left(\frac{|r|}{C} - \log\left(1 + |r|/C\right)\right) \quad \textbf{15}$$

It is extended to multidimensional case in a manner similar to (14) with the exception that

$$d_{ik} = \sum_{j=1}^{s} \left| x_{kj} - v_{ij} \right| \quad (\textbf{16})$$

and

$$J_i = \sum_{k=1}^{N} u_{ik}^m \rho(d_{ik}), i = 1,...,c \quad (17)$$

The plots of the 'fair' function for the unidimensional case and the plot of equation (14) with 'fair' function as the loss function would depict their convexity and unicity in the same vein as plots 2a and 2b. These plots are not plotted henceforth for the sake of brevity. The constant C is taken as 1.3998 for 0.95% asymptotic efficiency on the standard normal distribution [19].

Modifying the "Fair" estimator:

However it is to be noted that the extension to multidimensional case here is not done on the lines of (16) and (17) for the minimization of (17) involves solving coupled non-linear equations, the Jacobian of which while using the Newton Raphson's always vanishes. Solving through bisection involves searching the whole of the s dimensional space of $v_i = (v_{i1}, v_{i2}, ..., v_{is})$ for each $i = 1,...,c$ and is extremely intensive. Hence the extension was done through $d_{ik} = \|x_k - v_i\|_2 = \left[\sum_{j=1}^{s}(x_{kj} - v_{ij})^2\right]^{0.5}$. The graph of (17) with this modified distance metric is shown in figure 3c and is once again convex.

2.2.3 The Huber's Estimator [23]:

Loss function:

$$\rho(r) = \begin{cases} r^2/2\,; |r| \le k \\ k(|r| - k/2); \quad |r| \ge k \end{cases} \qquad \textbf{(18)}$$

The extension to multidimensional case is of the form

$$\rho(d_{ik}) = \begin{cases} \|x_k - v_i\|_2^2/2 \quad ; \|x_k - v_i\|_1^1 \le k \\ k\left(\|x_k - v_i\|_1^1 - k/2\right) \; ; \|x_k - v_i\|_1^1 \ge k \end{cases} \qquad \textbf{(19)}$$

The graphs for the form of (18) and when the loss function of (19) would again be convex as in plots 2a and 2b. The 95% asymptotic efficiency on the standard normal distribution is obtained for k=1.345 [19].

2.2.4 The Unsquared Euclidean

Loss function for unidimensional case is given by $\rho(r) = r$ and extension to the multidimensional case is of the form $\rho(d_{ik}) = \|x_k - v_i\|_2$. The loss function is very similar to the standard objective functional of the FCM in equation (7) except that the L_2 norm is left unsquared here whereas in equation (7) the squared L_2 norm is used. Thus the datum's influence on the estimate does not increase linearly with the residual error lending some robustness. In other words the influence function here is bounded unlike the case of squared Euclidean metric.

2.2.5 Cauchy's Estimator [19]:

The loss function for single dimension is $\rho(r) = \frac{c^2}{2}\log\left(1 + (r/c)^2\right)$ with an extension to multidimensions in a manner similar to (13). The recommended value of the tuning constant is c=2.3849.

3 Simulations with Synthetic Data

The objective here is to essentially investigate the performance of the family of FCM algorithms obtained by incorporating the various M-estimators and distance measures listed in the previous chapter when outliers are present. For this purpose a data set consisting of three primary clusters, each cluster containing 40 points as shown in figure 3 is employed. Outliers are introduced at the two positions shown in figure 3 marked by arrows. They are evenly distributed at the two positions within a square of 3X3 pixels centered at those two locations. The initial values for the fuzzy partition matrix and the cluster centers are initialized to the same random values across all the combinations of the FCM algorithm obtained by adopting the various M-estimators and distance measures for a given set of outliers. The results of simulation are tabulated in table 1. The left most column of the table indicates the number of outliers introduced in figure 3. The columns other than the leftmost signify the particular norm or M-estimator incorporated in the FCM objective functional and the entries in these denote the deviations of the computed centers from actual ones based on the Frobenius norm for that measure. The simulations were run with the fuzzy index $m = 2$. The iterations were stopped as soon as the sum square error between the locations occupied by the centers over two successive iterations falls less than 0.01. In other words the termination condition is $\sum_{i=1}^{c} \left\| v_i^n - v_i^{n-1} \right\|_2^2 \le 0.01$.

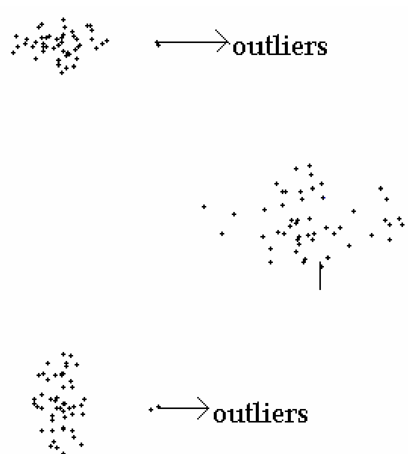


Fig. 3. Dataset used for testing. Outliers added equally at places shown by arrows

The simulations suggest that the L_1 norm and the Huber's M-estimator to be least sensitive to outliers and the L_2 norm to be the most vulnerable. The $L_{1.2}$ norm comes next. The $L_1 - L_2$, "fair" and the unsquared Euclidean estimators come next and can be grouped together for there does not seem to be an appreciable between them. They are far more robust when compared with L_p estimators for values of $p \geq 1.4$. The results of simulation tallies with the simulations reported by the authors in [1] for the L_p estimators in the sense the sensitivity to outliers increases with a corresponding increase in the value of p for the initial 40 outliers. The deviations from the actual centers with an increase in number of outliers can also be seen in general except for unsquared Euclidean and L_1 estimators for the specific case of 12 outliers which show a trifle higher deviation when compared with the case of 16 outliers.

In the next set of simulations the initial values of partition matrix are initialized using a hard partition that correctly partitions the three clusters and groups each outlier with its nearest cluster. The objective here was to probe whether running the robust algorithms over a initial partition obtained from a hard clustering algorithm gives better results when compared with those tabulated in table 1. Another objective was to explore into the possibility of any other M-estimator that otherwise do not work well with random starting conditions (Cauchy, Welsch [24] and Andrews [25]) would provide more robust estimates than L_2 if the initial conditions were more likely estimates of the partition. The results tabulated in table 2 however belie such expectations. The performance of the estimators of table 1 does not show any tangible improvement in table 2. Amongst the M-estimators that did not work with random initial conditions only the Cauchy's estimator appears to give acceptable results with the present set of initial conditions obtained from a correct hard partition. The estimates of Cauchy's are however not in any way more robust than the L_2 estimates as seen from table 2.

Table 1. Deviations of the actual centers from their estimates based on Frobenius norm. The initial partitions are randomly assigned but have same values across a particular row.

Outliers	Lp (p=1)	p=1.2	p=1.5	p=2	Unsquared Euclidean	L1-L2	Fair	Huber
0	0.464	0.493	0.876	1.232	1.014	0.982	0.913	0.487
4	1.125	1.167	1.778	3.335	2.167	2.165	1.130	2.04
8	1.360	1.384	2.583	6.197	2.35	2.687	1.953	2.36
12	2.395	3.150	5.867	10.25	4.44	2.997	2.608	2.395
16	2.683	2.904	6.314	13.36	3.162	3.613	3.504	2.504
20	2.621	3.742	6.839	14.141	3.57	3.722	4.557	2.621
30	2.998	4.327	8.895	17.931	5.293	6.153	6.201	2.92
40	4.794	6.943	21.744	23.494	10.513	7.863	9.217	4.813

Table 2. Deviations of the actual centers from their estimates based on Frobenius norm. The initial partitions are those obtained from a HCM algorithm

Outliers (total no)	AFC (Standard form)				AFC with Fair				AFC with L1-L2			
	Min Err	Max Err	Ave Err	BD %	Min Err	Max Err	Ave Err	BD %	Min Err	Max Err	Ave Err	BD%
0	0.755	3.257	1.13	0%	1.106	2.94	1.48	0%	1.233	3.178	1.516	0%
15	9.112	35.98	19.8	38%	10.102	10.10	10.1	0%	10.35	18.085	11.12	8%
30	11.50	44.75	31.4	57%	13.208	51.94	20.9	18%	13.65	54.12	21.59	18%
45	14.38	52.60	29.6	44%	15.127	53.52	26.6	26%	14.87	54.72	36.7	56%
60	17.43	53.51	39.0	58%	19.14	54.91	37.6	54%	19.42	55.36	37.82	58%

4. A Robotic Application

Robotic navigation includes sensing, scene interpretation and collision avoidance of stationary and moving objects in the robot's neighborhood. Often a robot is expected to navigate in unknown environments and hence needs to come to an understanding of the environment from its sensor readings. When the robot employs range sensors for probing the environment, preliminary processing of sensory data results in a point cloud representation of the environment as shown in figures 6a and 6b. Figure 6a shows an environment with 5 objects along with a navigating robot circular in shape and 6b its point cloud representation obtained by projecting raw sensor data onto a global reference frame. Operations such as extracting the endpoints and centers of the objects from the point cloud representations becomes mandatory in situations that demand object tracking, motion prediction and maneuvering past moving objects [26]. They also become essential for classifying objects as stationary or moving from the characteristics of the point cloud cluster that represents the objects [27]. Evidently the nature of the point cloud representation points to a clustering based solution for extracting the object centers and its further attributes such as the covariance matrix, eigenvalues and eigenvectors. Considering the predominantly linear shape of the clusters, the AFC algorithm that is specifically recommended for discerning flat (linear) clusters has been employed. Since sensor data is in general error prone and outlier ridden due to various grounds [28] need for robust clustering algorithms appears inevitable.

4.1 Integrating M-Estimators with the AFC Algorithm

The M-estimators are integrated with the AFC in a manner similar to that discussed in section 2.2. The objective functional of the AFC is written as $J(U,V)=\sum_{i=1}^{c}\sum_{k=1}^{N}u_{ik}^{m}\rho\left(d_{ik}^{2}\right)$ where $d_{ik}^{2}=\sum_{j=1}^{s}a_{ij}\left((\vec{x}_{kj}-\vec{v}_{ij})\bullet\vec{e}_{ij}\right)^{2}$ is the distance measure used in AFC algorithm for detecting flat clusters. Here e_{ij} is the j th unit eigenvector for the i th cluster and $a_{ij}=\lambda_{is}/\lambda_{ij}$ is the ratio of the smallest eigen value, λ_{is}, to the j th eigenvalue λ_{ij} for the i th cluster. Instead of solving for the prototypes by numerical approximations the method of W-estimator (equation 6) is used for obtaining their estimates. Despite the error that obviously creeps in when the weight function appearing in the right hand side of equation itself is a function of the prototype that is being estimated, the formulation of (6) is preferred considering the real time urgency of the situation in a robotic application. The standard AFC and the AFC modified by the L_1-L_2 and “fair” estimators where used for the comparative analysis. The centers are computed as

$$\vec{v}_i=\frac{\sum_{k=1}^{N}u_{ik}^{m}\psi(d_{ik}^{2})x_k}{\sum_{k=1}^{N}u_{ik}^{m}\psi(d_{ik}^{2})}\quad ;i=1,2,\ldots,c \qquad \mathbf{(20)}$$

where $\psi\left(d_{ik}^{2}\right)$ is the influence function and takes the value of unity, $1/2\left(\sqrt{1+d_{ik}^{2}/2}\right)$ and $C\left(1-\frac{1}{1+\sqrt{d_{ik}^{2}}/C}\right)\frac{1}{2\sqrt{d_{ik}^{2}}}$ for the standard, L_1-L_2 and the “fair” forms of the AFC algorithm respectively. Here d_{ik}^{2} is computed by taking the prototypes obtained in the previous iteration.

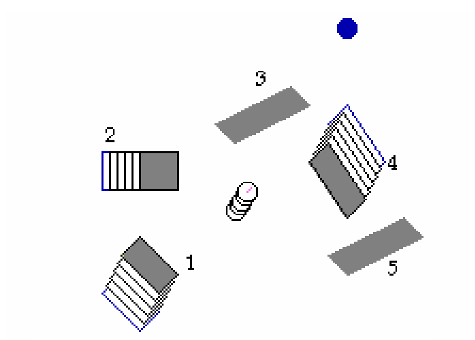


Fig 4a: Navigation amidst 3 static and 2 moving obstacles labeled one to five.

Figure 4b: Point cloud representation of range data of figure 4a obtained by projecting it onto a map

4.2 Simulation Results

Simulations are run on the sensory data obtained by navigating a robot in the environment of figure 7a. The environment contains 4 stationary objects. Two transient objects initially at positions ‘a’ and ‘b’ marked with arrows start moving away from the robot. Range data corresponding to these two transient objects appear as noise at locations ‘a’ and ‘b’ in the point cloud representation shown in figure 7b and can hinder accurate extraction of the objects’ features. Another source of noise is due to sensor misreadings at the corner formed at the meeting point of the two objects marked as ‘c’ in figure 3a. At the corners due to the phenomenon of multiple reflections [28] the range readings at times are higher than the usual value and appear as outliers at location ‘c’ in figure 3b. Thus the simulation environment of figure 3a has three sources of outliers and the AFC algorithm is run over varying number of outliers. The results are tabulated for 5, 10, 15 and 20 outliers at each of the three locations. The number of data points for each of the object was limited at 40. The results are tabulated in table 3 for the standard AFC and the AFC modified with the $L_1 - L_2$ and “fair” estimators.

A number of runs were performed for each algorithm with random initializations for a given number of outliers. The minimum, maximum and the average error over these runs measured through the Frobenius norm distance for a given number of outliers are reported as 3 columns for each version of AFC in table 3. The breakdown percentage (BP) appears as the fourth column for that version. By breakdown we mean informally that the estimate of the prototype during that run has become completely unreliable. More formal definitions of break down from a statistical perspective can be obtained from [29]. The table suggests that while the minimum error for the standard AFC is marginally ahead of the other two AFC versions the percentage breakdown is far less for the AFC modified by the M-estimators when compared with the standard version for 5 and 10 numbers of outliers per location. Thereby the average error for the modified forms of AFC are also low when compared with the standard version. However as the number of outliers increases the fidelity of all the three versions decrease sharply and are rendered unreliable. It is to be noted that the outliers are placed in such a position that the algorithms can be tricked to form an erroneous cluster with the cluster’s center settling right in the middle between the two locations of outliers. This phenomenon is shown in figure 7c with the erroneous cluster shown by a line and its center indicated through an arrow. The propensity for the algorithms to fall into this trap increases as the number of outliers at that location increases that serves as an attractor or sink for one of the four prototypes to be estimated. The authors feel that the high average frobenius error with increase in outliers across all algorithms may be due to this factor.

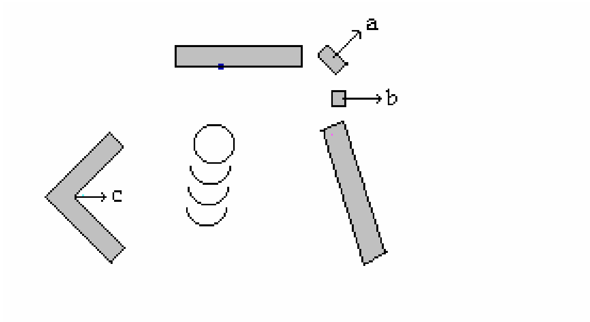


Fig. 5a. Simulation environment used for testing. Robot is shown in circles. 'a' and 'b' are transient objects that move away. 'c' is a corner where sensor errors are prone to occur

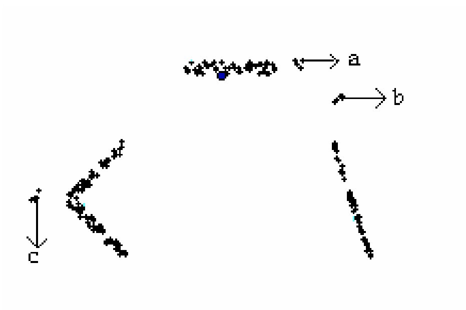


Fig 5b: Transient objects of figure 5a appear as outliers at 'a' and 'b'. Multiple reflections at corner 'c' of 5a appear as outliers at location 'c' in this figure

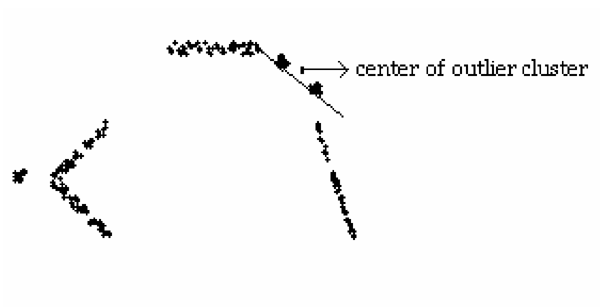


Fig 5c: The tendency to form a line like cluster between the two sources of outliers increases as the outlier concentration increases. The center of the outlier cluster is indicated by an arrow

Table 3. Maximum, minimum and average error for the AFC versions used for partitioning the sensor data. The breakdown percentages (BD%) are also indicated

Outliers (total no)	AFC (Standard form)				AFC with Fair				AFC with L1-L2			
	Min Err	Max Err	Ave Err	BD%	Min Err	Max Err	Ave Err	BD%	Min Err	Max Err	Ave Err	BD%
0	0.755	3.257	1.134	0%	1.106	2.944	1.487	0%	1.233	3.178	1.516	0%
15	9.112	35.984	19.86	38%	10.102	10.108	10.104	0%	10.358	18.085	11.12	8%
30	11.504	44.757	31.45	57%	13.208	51.944	20.95	18%	13.65	54.12	21.59	18%
45	14.38	52.60	29.6	44%	15.127	53.52	26.6	26%	14.87	54.72	36.7	56%
60	17.431	53.51	39.07	58%	19.14	54.91	37.69	54%	19.42	55.36	37.82	58%

5 Concluding Remarks

Researchers have investigated the robustification of fuzzy clustering algorithms. The Fuzzy-C-Medians (FCMED) had been proposed as a robust alternative to the FCM [12] and an extension of FCM over generalized L_p distances was recently formulated [1]. On the other hand the extension of the FCM over some M-estimators of robust statistics was analyzed by Krishnapuram and others [14,16,18]. In this paper the FCM is further extended over the $L_1 - L_2$, "fair", unsquared Euclidean, Cauchy and Huber's estimators the extensions over which do not seem to have been reported in literature. The performance of these modified versions of FCM under the duress of outliers has been compared with the performance of the previous extensions of FCM over L_p norm distances. The results indicate that the Huber's estimator emerges on par with the L_1 estimator as being the least vulnerable to outliers. The $L_{1.2}$ estimator comes next closely followed by a group of three M-estimators namely, "Fair", $L_1 - L_2$ and the unsquared Euclidean, which gives similar performance. This group of three estimators is far more robust than L_p estimates for $p \geq 1.4$. The L_2 and Cauchy estimators fill up the bottom ranks of the hierarchy. Moreover the performance of Cauchy's estimator is reliable only when the initial values of the partition matrix are obtained from a hard partition that correctly partitions the data points including the outliers to its respective cluster.

As a possible application the AFC algorithm was also extended over the $L_1 - L_2$ and "Fair" estimators and compared on range data acquired by the sensors of the navigating robot. Results seem to indicate that the $L_1 - L_2$ and "Fair" forms of the AFC algorithm are less prone to breakdown when compared with the standard AFC version when the number of outliers is moderate. However as the number of outliers increase the performance of the modified versions of the FCM falls and there appears not much to chose amongst the various versions. The authors feel that making use of the form of equation (20) for obtaining an analytical expression for cluster prototypes has built in it some measure of error that could lessen the performance of the robust versions of AFC. In that sense the cluster prototypes estimated through numerical techniques as applied on the extensions of FCM were free from errors that arise when equation (20) is employed to estimate the prototypes. However the form of equation (20), which bypasses the need for numerical iterations was employed considering the real-time nature of robotic application. It is felt here that further investigations are required before a categorical ordering of the robustness in M-estimators on AFC can be made as it could be made for the various estimates on FCM such as $L_1 \approx$ .Huber > $L_{1.2}$ > $L_1 - L_2 \approx$ Fair $\approx$ Unsquared-Euclidean > $L_p(p \geq 1.4) > L_2 \approx$ Cauchy.

Reference:

1. J. Bezdek et. al., "Generalized fuzzy-C-means clustering strategies using Lp norm distances", *IEEE Trans. Fuzzy Sys*. Vol. 8, no. 5, Oct. 2000
2. R. Krishnapuram and J. Kim, “A Note on the Gustafson-Kessel and Adaptive fuzzy clustering algorithms”, *IEEE Trans. on Fuzzy Systems*, vol. 7, no. 4, Aug. 1999
3. R. J. Boscovich, in *Bonoinensi Scientarium et Artum Instuto Atque Academia Commentari* 4, 353-396, Ch. 2, 1997
4 D. Berouilli, in *Acta Academiae Scientiorum Petropolitanae*, 1, 3-33. English translation by C. G. Alen in *Biometrika* vol. 48, pp 3-13, 1961
5 T. S. Ferguson, “On the rejection of outliers”, *Proc. of the fourth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 253-287
6. P. Rousseouw and G Bassett, "The remedian: A robust averaging method for large data sets, ", *J. Amer. Assoc*., vol. 85, no. 409, pp. 97-104. Mar. 1990
7. K. Ogata, "Discrete time control systems", Prentice Hall Publications, NJ, 1987.
8. E. Dougherty and J. Astola, "An introduction to non-linear image processing" Bellingham, WA: SPIE Opt. Engrg. 1994
9. D. Bikes and Y. Dodge, "Alternative methods of regression", John Wiley, NY, 1993.
10 J. Bezdek, "Pattern Recognition with Fuzzy Objective Function", Plenum Press, NY, 1981.
11. J.C. Bezdek et. al, "Fuzzy models and algorithms for pattern recognition and image processing", Norwell, MA: Kluwer, 1999
12. P. R. Kersten, "Implementing the fuzzy -C- medians algorithm", in Proc. *IEEE Int'l Conf. on Fuzzy Sys.*, pp. 957-962, Barcelona, 1997
13. --------------, "Fuzzy order statistics and their applications to fuzzy clustering", *IEEE Trans. Fuzzy Sys.*, vol. 7, pp 708-712, 1999.
14 H. Frigui and R. Krishnapuram, “A Robust Clustering Algorithm Based on M-estimator”, Proc. First Int’l Conf. on Neural, Parallel and Scientific Computations, vol. 1, pp. 163-116, Atlanta, May 1995.
15 P. J. Huber, “Robust Statistics”. New York: John Wiley and Sons: 1981
16. J. Kim, R. Krishnapuram and R. Dave, “On Robustifying the C-Means Algorithm”, *Proc. ISUMA/NAFIPS*, pp. 630-635, Md. 1995.
17 P.J. Rosseeuw and A. M. Leroy, “Robust Regression and Outlier Detection” New York: John Wiley and Sons: 1987
18 R. N. Dave and R. Krishnapuram, “Robust Clustering Methods: A Unified View”, *IEEE Trans. on Fuzzy Systems*, vol. 5, no. 2, May 1997
19 W. J. Rey, “Introduction to Robust and Quasi-Robust Statistical Methods”. Springer, Berlin, Hiedelberg, 1983.
20 http://www-sop.inria.fr/robotvis/personnel/zzhang/Publis/Tutorial-Estim/node24.html

21. K V Rosepack – Robust Estimation Package, *ACM Signum letter*, vol. 13, pp. 18-19, 1978.

22 D. Coleman et. al. “A System of subroutines for iteratively reweighted least squares computations”, *ACM. Trans on Math. Software*, vol. 6, pp. 327-336, 1980.
23 P. J. Huber, “Robust estimation of a location parameter”, *Ann. Math. Statist*, vol. 35, pp. 73-101, 1964.
24 J. E. Dennis and R. E. Welsch, “Techniques for non-linear least squares and regression”, *Proc. Amer. Statist. Ass.*, pp.83-87, 1976.
25 D. F. Andrews, “A robust method for multiple linear regression” *Technometrics*, vol. 16, pp. 524-531, 1977.
26. K M Krishna and P K Kalra, “Detecting Tracking and Avoidance of Multiple Dynamic Objects”, *Journal of Intelligent and Robotic Systems*, 33: 371-408
27 K M Krishna and P K Kalra, “When does the robot perceive a dynamic object”, *to appear in Journal of Robotic Systems*, 19(2): Feb 2002.
28 J. Borenstein and Y Koren, "Real-time obstacle avoidance for fast mobile robots", *IEEE Trans. SMC*, vol 19, no 5, Sep/Oct. 1989

APPENDIX (THE FCM ALGORITHM)

A large family of fuzzy clustering algorithms is based on the minimization of the *fuzzy-c-means* functional formulated as [10

$$J(Z,U,V) = \sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m \left\| z_k - v_i \right\|^2 \qquad \textbf{(15)}$$

where Z is the data set consisting of feature vectors z_k, $k = \{0,1,..,N\}$ and $U = [u_{ik}] \in [0,1]$ is the fuzzy partition matrix [10of Z is subject to the following conditions:

$$\sum_{i=1}^{c} u_{ik} = 1, \quad 1 \le k \le N \qquad \textbf{(16)}$$

and

$$0 \le \sum_{k=1}^{N} u_{ik} \le N, 1 \le i \le c \qquad \textbf{(17)}$$

The vector $V = [v_1, v_2, ..., v_c], v_i \in R^n$ is a cluster of prototypes (centers), which have to be determined and $D_{ik} = \left\| z_k - v_i \right\|^2$ is the squared inner product norm (typically Euclidean) and $m \in [1, \infty)$ is a weighting exponent that determines the fuzziness of the resulting clusters and is usually set to 2.

The stationary points of the objective function can be found by adjoining the constraint of 17 and 16 by means of Lagrange multipliers:

$$J(Z;U,V,\lambda)=\sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^{m}\left\|z_k-v_i\right\|^2+\sum_{k=1}^{N}\lambda_k\left(\sum_{i=1}^{c}\mu_{ik}-1\right) \qquad \textbf{(18)}$$

Setting the gradients of J with respect to U,V and λ to zero locates the minimum value of 18 at:

$$\mu_{ik}=\frac{1}{\sum_{j=1}^{c}(D_{ik}/D_{jk})^{2/(m-1)}} \qquad \textbf{(19)}$$

and

$$v_i=\frac{\sum_{k=1}^{N}(\mu_{ik})^{m}z_k}{\sum_{k=1}^{N}(\mu_{ik})^{m}} \qquad \textbf{(20)}$$

The algorithm converges to its optimal fuzzy partition matrix and cluster centers through a Picard iterative procedure till the difference in the norm between successive partition matrices is less than a prescribed tolerance ε.