Implementation of Possibilistic Fuzzy C-Means Clustering Algorithm in Matlab

Neelam Kumari, Bhawna Sharma, Dr. Deepti Gaur
Dept.of Computer Science &IT,ITMUniversity,Gurgaon, INDIA.
singh.neelam693@gmail.com
bhawnash.6@gmail.com
deepti_k_Gaur@yahoo.com

***Abstract*—**Clustering means classifying the given observation data sets into subgroups or clusters. It is a process of grouping data objects into disjointed clusters so that the data in the same cluster are similar, yet data belonging to different clusters are different. Different fuzzy data clustering algorithms exist such as Fuzzy C-Means(FCM), Possibilistic C-Means(PCM), Fuzzy Possibilistic C-Means(FPCM) and Possibilistic Fuzzy C-Means(PFCM). In this paper we present the implementation of PFCM algorithm in Matlab and we test the algorithm on two different data sets.

***Index Terms*—** Data clustering , Clustering algorithms, K-Means, FCM, PCM, FPCM, PFCM.

I.INTRODUCTION

There are various algorithms of data clustering, every algorithm has its own advantages and disadvantages. Such as K-Means algorithm does not allow overlapping of clusters which is a major disadvantage of this algorithm. The FCM has problems dealing with noise and outliers. The PCM has the problem of coincident clusters and the FPCM has difficulties when the data set is big because the typicality values will be very small. All the apparent problem of FPCM is that it imposes a constraint on the typicality values (sum of the typicalities over all data points to a particular cluster is 1).We relax the constraint on the typicality values but retain the column constraint on the membership values. The PFCM is a good clustering algorithm to perform classification tests because it possesses capabilities to give more importance to typicalities or membership values. PFCM is a hybridization of PCM and FCM that often avoids various problems of PCM,FCM and FPCM.

II. PFCM CLUSTERING ALGORITHM

PFCM lead to optimize the following objective function:

$$min\left\{J_{m,\eta}(U,T,V;X) = \sum_{k=1}^{n}\sum_{i=1}^{c}\left(au_{ik}^{m}+bt_{ik}^{\eta}\right)\times\|x_k - v_i\|_A^2 + \sum_{i=1}^{c}\gamma_i\sum_{k=1}^{n}(1-t_{ik})^{\eta}\right\} \quad (1)$$

subject to the constraints $\sum_{i=1}^{c} u_{ik} = 1\ \forall\ k\ , and\ 0 \le u_{ik}\ , t_{ik} \le 1$.Here a > 0, b > 0, m > 1,$\eta > 1$ and $J_{m,\eta}$is the objective function. U is the partition matrix. T is the typicality matrix. V is a vector of cluster centers, X is a set of all data points, x represents a data point, n is the number of data points and c is the number of cluster centers which are described by s coordinates.$\|x_k - v_i\|_A$ is any norm used to calculate the distance between i^{th}cluster center and k^{th} data set(also represented by D_{ikA}). Here we are using Euclidean Distance(calculated by using Equation 4).

The constants a and b define the relative importance of fuzzy membership and typicality values in the objective function. Note that, in Eq. 1 u_{ik}has the same meaning of membership as that in FCM. Similarly,t_{ik}has the same interpretation of typicality as in PCM. If we increase the importance (weight) of membership then that necessarily forces us to reduce the importance of typicality by the same amount. Also, we will see later that the optimal typicality values depend on the magnitude of b. So by constraining a $+$ b $= 1$, we lose modeling

flexibility.PFCM uses the objective functions of PCM and FCM given in Eq.2 and Eq.3 respectively.

$$\min\left\{J_m(U,V;X) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m\|x_k - v_i\|_A^2\right\} \quad (2)$$

$$min\left\{P_m(T,V;X,\gamma) = \sum_{k=1}^{n}\sum_{i=1}^{c}(t_{ik})^m\|x_k - v_i\|_A^2 + \sum_{i=1}^{c}\gamma_i\sum_{k=1}^{n}(1-t_{ik})^m\right\} \quad (3)$$

$$D_{ikA} = \left[\sum_{j=1}^{s}(x_{kj} - v_{ij})^2\right]^{1/2} \quad (4)$$

If b = 0, and $\gamma_i = 0$ for all i , then Eq.1 reduces to the FCM optimization problem in Eq.2; while converts it to the usual PCM model in Eq.3. Later, we will see that when b = 0,even if we do not set $\gamma_i = 0$ for all i ,Equation 1 implicitly becomes equivalent to the FCM model. Like FPCM, under the usual conditions placed on c-means optimization problems, we get the first-order necessary conditions for extrema of $J_{m,\eta}$.

$$u_{ik} = \left(\sum_{j=1}^{c}\left(\frac{D_{ikA}}{D_{jkA}}\right)^{2/(m-1)}\right)^{-1} \quad (5)$$
$$1 \le i \le c;\ 1 \le k \le n$$

$$t_{ik} = \frac{1}{1+\left(\frac{b}{\gamma_i}D_{ikA}^2\right)^{1/(\eta-1)}} \quad (6)$$
$$1 \le i \le c;\ 1 \le k \le n$$

$$v_i = \frac{\sum_{k=1}^{n}(au_{ik}^m + bt_{ik}^\eta)x^k}{\sum_{k=1}^{n}(au_{ik}^m + bt_{ik}^\eta)} \quad (7)$$
$$1 \le i \le c$$

PFCM behaves like FCM as the exponents grow without bound. That is, irrespective of the values of the constants a and b, all c centroids approach the overall (grand) mean as $m \to \infty$ and $\eta \to \infty$ .Equation 7 shows that if we use a high value of b compared to a , then the centroids will be more influenced by the typicality values than the membership values. On the other hand, if we use a higher value of a then the centroids will be more influenced by the membership values. Thus, to reduce the effect of outliers, we should use a higher value for b than a. Similar effects can also be obtained by controlling the choice of η . For example, if we use a large value of m and a smaller value for η , then the effect of outliers on the centroids will be reduced. However, a very large value of m will reduce the effect of memberships on the prototypes and the model will behave more like the PCM model.

III. EXPERIMENTAL ANALYSIS

We run the algorithm for the two data sets. The Matlab code for following six files for Possiblistic Fuzzy C-Means clustering algorithm are as follows:

pfcm.m

```
function[center,U,T,obj_fcn]=
pfcm(data,cluster_n,options)

%pfcmDatasetclusteringusingpossiblistic fuzzy c-
%meansclustering.

[CENTER,U,OBJ_FCN]=FCM(DATA,N_CLUST
ER)    %findsN_CLUSTER

%number of clusters in the data set DATA.
%DATA is size M-by-N, where M is the number
%of data points and N is the number of coordinates
%for each datapoint. The coordinates for each
%cluster center are returned in the rows of the
%matrix CENTER. The membership function
%matrix U contains the grade of membership of
%each data point in each cluster. The values 0 and
%1 indicate no membership and full membership
%respectively. Grades between 0 and 1 indicate
%that the data point has partial membership in a
%cluster. At each iteration, a possiblistic degree
```

```
%same as in pcm T objective function is
%minimized to find the best location for the
%clusters and its values are returned in OBJ_FCN.

%[CENTER,..]=PFCM(DATA,N_CLUSTER,OPTIO
%NS)specifies a vector of options fort he

%clustering process:

% OPTIONS(1): exponent for the matrixUdefault:
%2.0)

%  OPTIONS(2): maximum number of % iterations
%(default: 100)

%   OPTIONS(3): minimum amount of

% improvement         (default: 1e-5)

%   OPTIONS(4): info display during

% iteration         (default: 1)

%   OPTIONS(5): user defind constant a
%(default: 1)

%   OPTIONS(6): user defined constant b %should
be greater than a  (default:4)

%   OPTIONS(7):  user defined constant%nc
(default:nc=2.0)

% The clustering process stops when the maximum
%  number of iterations is reached, or when
%objective function improvement between two

% consecutiveiterations is less than the minimum

% amount specified. Use NaN to select the default

% value.See also pinitf, tinitf, pdistfcm,pstepfcm

ifnargin ~= 2 &nargin ~= 3,

error('Too many or too few input arguments!');end

data_n = size(data, 1);

in_n = size(data, 2);

% Change the following to set default options

default_options = [2;100;1e-;1;1;4;2];

ifnargin == 2,options = default_options;

else

% If "options" is not fully specified, pad it with

% default values.

if length(options) < 7,

tmp = default_options;

tmp(1:length(options)) = options;options = tmp;

end

% If some entries of "options" are nan's, replace them
% with defaults.

nan_index = find(isnan(options)==1);

options(nan_index) = default_options(nan_index);

if options(1) <= 1,

error('The exponent should be greater than 1!');

end

end

expo = options(1);

max_iter = options(2);

min_impro = options(3);

display = options(4);

a=options(5);

b=options(6);

nc =options(7);

ni=input('enter value of ni');

obj_fcn = zeros(max_iter, 1);

% Array for objective function

center = input('initialize the centre');
```

```
U = pinitf(cluster_n, data_n); % Initial fuzzy partition

T = tinitf(cluster_n,data_n);  % Initial typicality
matrix
for i = 1:max_iter,

   [U,T,center, obj_fcn(i)] =
pstepfcm(data,center,U,T,cluster_n, expo,a,b,nc,ni);

if display,

fprintf('Iteration count = %d, obj. fcn = %f\n', i,
obj_fcn(i));

end

% check termination condition

if i > 1,

if abs(obj_fcn(i) - obj_fcn(i-1)) <min_impro,break;
end,endenditer_n = i; % Actual number of terations

obj_fcn(iter_n+1:max_iter) = [];
```

tinitf.m

```
function T = pinitf(cluster_n, data_n)

T = rand(cluster_n, data_n);

col_sum = sum(T);

u=col_sum(ones(cluster_n, 1), :);

 T = T./u;
```

pinitf.m

```
function U = pinitf(cluster_n, data_n)

U = rand(cluster_n, data_n);

col_sum = sum(U);

u=col_sum(ones(cluster_n, 1), :);

U = U./u;
```

pstepfcm.m

```
function [U_new,T_new,center_new,obj_fcn]=
pstepfcm(data,center,U,T,cluster_n, expo,a,b,nc,ni)

% PSTEPFCM One step in possiblisticfuzzy c-mean
% clusteringperforms one iteration of pfcm

% clustering, where DATA: matrix of data to be

% clustered. (Each row is a data point.)U:partition

% matrix. (U(i,j) is the MF value of data j in cluster
% j.) CLUSTER_N: number of clusters.EXPO:

% exponent (> 1) for the partition matrix.U_NEW:
%new partition  matrix of fcmT_new: new partition
%matrix of pcmCENTER: center of clusters. (Each
% row is a center.)  obj_fcn: objective function for
% partition U and T. Note that the situation of

 % "singularity" (one of the data points isexactly the
% same as one of the cluster centers)

% isnotchecked.However, it hardly occurs in

% practice.

mf = U.^expo;  % MF matrix after exponential
%modification

tf=T.^nc;tfo=(1-T).^nc;

center_new = (a.*mf+b.*tf)*data./((ones(size(data,
2), 1)*sum(a.*mf'+b.*tf'))')

dist = distfcm(center, data)

% fill the distance matrix

obj_fcn=sum(sum((dist.^2).*(a.*mf+b.*tf)))+sum(ni
*sum(tfo))

 % objective function

tmp = dist.^(-2/(expo-1));

U_new=tmp./(ones(cluster_n,1)*sum(tmp));

tmpt=((b/ni)*dist.^2).^(1/(nc-1));

T_new = 1./(1+tmpt);
```

pdistfcm.m

```
function out = pdistfcm(center, data)
```

```
%PDISTFCM Distance measure in possiblistic fuzzy
%c-mean clustering.OUT = PDISTFCM(CENTER,
%DATA) calculates the Euclidean distancebetween
%each row in CENTER and each row in DATA, and
%returns a distance matrix OUT of size M by N,
%where M and N are row  dimensions of CENTER
%and DATA, respectively, and OUT(I, J) isthe
%distance between CENTER(I,:) and DATA(J,:).

out = zeros(size(center, 1), size(data, 1));

% fill the output matrix

if size(center, 2) > 1,

for k = 1:size(center, 1),

  out(k, :) = sqrt(sum(((data-ones(size(data,
1),1)*center(k,:)).^2)'));

end

else

for k = 1:size(center, 1),

  out(k, :) = abs(center(k)-data)';

end

end
```

centpfcm.m

```
%see also
%pfcm,dataset1,pinitf,tinitf,pdistfcm,pstepfcm.

[center,U,T,obj_fcn] = pfcm(dataset1,2);

plot(dataset1(:,1), dataset1(:,2),'o');

holdon;

maxU = max(U);

%Find data points with highest grade of membership
%in cluster1

index1 = find(U(1,:) == maxU);

% Find the data points with highest grade of
%membership in cluster2

index2 = find(U(2,:) == maxU);

line(dataset1(index1,1),dataset1(index1,2),'marker','*'
,'color','g);

line(dataset1(index2,1),dataset1(index2,2),'marker','*'
,'color','r');

%Plot the cluster centers

plot([center([1 2],1)],[center([1
2],2)],'*','color','k')

holdoff;
```

dataset1.dat(data set-1)

	1	2
1	-5.0	0.0
2	-3.34	1.67
3	-3.34	0.0
4	-3.34	-1.67
5	-1.67	0.0
6	1.67	0.0
7	3.34	1.67
8	3.34	0.0
9	3.34	-1.67
10	5.0	0.0
11	0.0	0.0
12	0.0	10.0

The file dataset1.dat stores the whole data set on which PFCM clustering algorithm is to be applied.The result obtained after calling pfcm.m function(implemented in matlab) for the above data set-1 and data Set-2 is shown below:

OUTPUT:

Centpfcm

center value of ni0.08
initialize the centre[0.07 0.36;0.40 0.99]

Iteration count = 1, obj. fcn = 747.953767

Iteration count = 2, obj. fcn = 115.889242

Iteration count = 3, obj. fcn = 114.865973

Iteration count = 4, obj. fcn = 112.480753

Iteration count = 5, obj. fcn = 106.975976

Iteration count = 6, obj. fcn = 97.427050

Iteration count = 7, obj. fcn = 87.573956

Iteration count = 8, obj. fcn = 82.246424

Iteration count = 9, obj. fcn = 80.609839

Iteration count = 10, obj. fcn = 80.231624

Iteration count = 11, obj. fcn = 80.145442

Iteration count = 12, obj. fcn = 80.123254

Iteration count = 13, obj. fcn = 80.116857

Iteration count = 14, obj. fcn = 80.114881

Iteration count = 15, obj. fcn = 80.114251

Iteration count = 16, obj. fcn = 80.114046

Iteration count = 17, obj. fcn = 80.113980

Iteration count = 18, obj. fcn = 80.113958

Iteration count = 19, obj. fcn = 80.113951

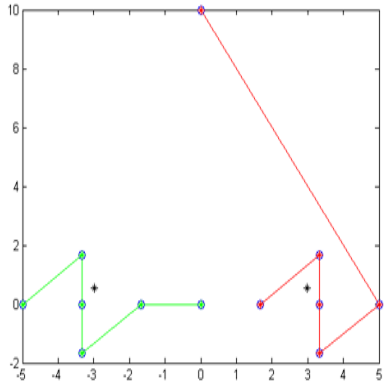

Figure-1

fcmdata.dat(Data Set-2)

	1	2
1	0.21895919	0.71971108
2	0.67929641	0.31389837
3	0.83096535	0.47903993
4	0.053461635	0.77931485
5	0.68677271	0.52219568
6	0.091964891	0.71405761
7	0.65391896	0.16631787
8	0.70119059	0.422224
9	0.91032083	0.31883553
10	0.73608188	0.45760685
11	0.63263857	0.14380738
12	0.7226604	0.3206678
13	0.75335583	0.46373323
14	0.072685883	0.87749311
15	0.27270997	0.64868126
16	0.76649478	0.39238853
17	0.35926498	0.64307502
18	0.90465309	0.18695199
19	0.49397668	0.54814877
20	0.26614451	0.73637975
21	0.073749075	0.79043769
22	0.52974739	0.40064069
23	0.46444582	0.56809172
24	0.77020455	0.16379344
25	0.62954342	0.23056262
26	0.73622451	0.065969527
27	0.88857221	0.23088594
28	0.5132737	0.31833422
29	0.59111358	0.24322982
30	0.53730398	0.38338957
31	0.46791737	0.62853425
32	0.28721237	0.7751988
33	0.1783277	0.7663085
34	0.80240573	0.25589577
35	0.49848012	0.33800649
36	0.55458385	0.51581954
37	0.89073748	0.37895436
38	0.62484929	0.085975014
39	0.71470997	0.503418
40	0.2399108	0.6709439
41	0.68134621	0.060274597
42	0.147533	0.87615123
43	0.58718662	0.47366216
44	0.59010861	0.48229278
45	0.55614614	0.15842396
46	0.40876669	0.67277244
47	0.56489868	0.42322414
48	0.48851455	0.24175423
49	0.65125374	0.54492761
50	0.24784176	0.91691429
51	0.4764318	0.57842135
52	0.38931417	0.52762005
53	0.20325033	0.71023354
54	0.94748678	0.33117965
55	0.13118853	0.75624968
56	0.88564837	0.28843555
57	0.09217363	0.73633707
58	0.36533903	0.54023723
59	0.25305736	0.76713302
60	0.78315317	0.50865692
61	0.34952414	0.73388084
62	0.21524838	0.86596086
63	0.67959237	0.20414434
64	0.25012559	0.82884477
65	0.86085984	0.3939987
66	0.81756148	0.49034177
67	0.75584353	0.17416348
68	0.82469739	0.36823673
69	0.10343393	0.79543708
70	0.57671664	0.30030901
71	0.87656572	0.29353873
72	0.44003866	0.50536909
73	0.86926374	0.28842873
74	0.88603112	0.13881256
75	0.46332273	0.5554808
76	0.71342232	0.47622826
77	0.66767907	0.24682623
78	0.68204912	0.4084118
79	0.31573241	0.68839481
80	0.46753178	0.53192527
81	0.31917759	0.88998057
82	0.68249423	0.48122673
83	0.83641988	0.14312388
84	0.70892061	0.48309189
85	0.82870795	0.32547527
86	0.2135468	0.65412826
87	0.38985359	0.6198854

88	0.77686582	0.30887354
89	0.7838652	0.23765611
90	0.28215589	0.92267568
91	0.81972609	0.16874402
92	0.60101011	0.27616354
93	0.82835472	0.50918299
94	0.15773118	0.83858058
95	0.23359919	0.72674889
96	0.63471744	0.33295071
97	0.79476981	0.45126372
98	0.69624281	0.38926233
99	0.75294041	0.33203456
100	0.66952064	0.50487466
101	0.63342991	0.42841739
102	0.22700772	0.84878311
103	0.69983444	0.26863618
104	0.52612328	0.47793654
105	0.32966639	0.75485767
106	0.48532518	0.40283149
107	0.8602257	0.38888944
108	0.55683578	0.22796971
109	0.73899661	0.48685151
110	0.52854827	0.30300425
111	0.31073887	0.59246226
112	0.58811912	0.51315049
113	0.51808359	0.52021714
114	0.37022628	0.72335304
115	0.47589622	0.29663054
116	0.078263208	0.76459728
117	0.36974201	0.58650577
118	0.67178415	0.21699766
119	0.67623692	0.079747635
120	0.51393637	0.31850888
121	0.72860836	0.17882267
122	0.72076782	0.472587
123	0.32156009	0.74857788
124	0.46043417	0.34848546
125	0.6613555	0.064064059
126	0.6056397	0.14123407
127	0.6700984	0.49318005
128	0.52280771	0.29752862
129	0.26661332	0.70293582
130	0.24673315	0.66149041
131	0.81710133	0.51471396
132	0.16074891	0.88552542
133	0.7078263	0.1353905
134	0.43663845	0.50815926
135	0.75171009	0.19508763
136	0.90330118	0.3487054
137	0.5847872	0.080508049
138	0.62686137	0.47778535
139	0.65905333	0.13837822
140	0.53866129	0.13050593

Centerpfcm.m

```
%See also
%fcmdata,pfcm,pinitf,tinitf,pdistfcm,pstepfcm.
loadfcmdata.dat
[center, U,T, obj_fcn] = pfcm(fcmdata, 2);
maxU = max(U);
index1 = find(U(1, :) == maxU);
index2 = find(U(2, :) == maxU);
line(fcmdata(index1, 1), fcmdata(index1, 2),
'linestyle',...'none','marker', 'o','color','g');
line(fcmdata(index2,1),fcmdata(index2,2),'linestyle',..
.'none','marker', 'x','color','r');
holdon
plot(center(1,1),center(1,2),'kx','markersize',15,'Line
Width',2)
plot(center(2,1),center(2,2),'kx','markersize',15,'Line
Width',2)
```

OUTPUT:

```
Centerpfcm

enter value of ni0.085

initialize the centre[0.07 0.36;0.40 0.99]

Iteration count = 1, obj. fcn = 169.719992

Iteration count = 2, obj. fcn = 30.906949

Iteration count = 3, obj. fcn = 28.911804

Iteration count = 4, obj. fcn = 28.624657

Iteration count = 5, obj. fcn = 30.545798

Iteration count = 6, obj. fcn = 30.565587

Iteration count = 7, obj. fcn = 32.824230

Iteration count = 8, obj. fcn = 32.436485

 Iteration count = 9, obj. fcn = 34.899085

Iteration count = 10, obj. fcn = 33.981948

 Iteration count = 11, obj. fcn = 36.337822

Iteration count = 12, obj. fcn = 34.968145

 Iteration count = 13, obj. fcn = 36.884404

Iteration count = 14, obj. fcn = 35.170103

Iteration count = 15, obj. fcn = 36.419926

Iteration count = 16, obj. fcn = 34.483684

Iteration count=17,obj.fcn=35.006450

Iteration count= 18,obj. fcn=33.015857
```

Iteration count = 19, obj. fcn = 32.888743

Iteration count = 20, obj. fcn = 31.018486

Iteration count = 21, obj. fcn = 30.451930

Iteration count = 22, obj. fcn = 28.824755

Iteration count = 23, obj. fcn = 28.097759

Iteration count = 24, obj. fcn = 26.787371

Iteration count = 25, obj. fcn = 26.062699

Iteration count = 27, obj. fcn = 24.435272

Iteration count = 28, obj. fcn = 23.771498

Iteration count = 29, obj. fcn = 23.218869

Iteration count = 30, obj. fcn = 22.764694

Iteration count = 31, obj. fcn = 22.360895

Iteration count = 32, obj. fcn = 22.057192

Iteration count = 33, obj. fcn = 21.795550

Iteration count = 34, obj. fcn = 21.606641

Iteration count = 35, obj. fcn = 21.456819

Iteration count = 36, obj. fcn = 21.351975

Iteration count = 37, obj. fcn = 21.275950

Iteration count = 38, obj. fcn = 21.224284

Iteration count = 39, obj. fcn = 21.189212

Iteration count = 40, obj. fcn = 21.165943

Iteration count = 41, obj. fcn = 21.150726

Iteration count = 42, obj. fcn = 21.140799

Iteration count = 43, obj. fcn = 21.134421

Iteration count = 44, obj. fcn = 21.130304

Iteration count = 45, obj. fcn = 21.127677

Iteration count = 46, obj. fcn = 21.125992

Iteration count = 47, obj. fcn = 21.124918

Iteration count = 48, obj. fcn = 21.124232

Iteration count = 49, obj. fcn = 21.123794

Iteration count = 50, obj. fcn = 21.123515

Iteration count = 51, obj. fcn = 21.123337

Iteration count = 52, obj. fcn = 21.123223

Iteration count = 53, obj. fcn = 21.123151

Iteration count = 54, obj. fcn = 21.123104

Iteration count = 55, obj. fcn = 21.123075

Iteration count = 56, obj. fcn = 21.123056

Iteration count = 57, obj. fcn = 21.123044

Iteration count = 58, obj. fcn = 21.123036

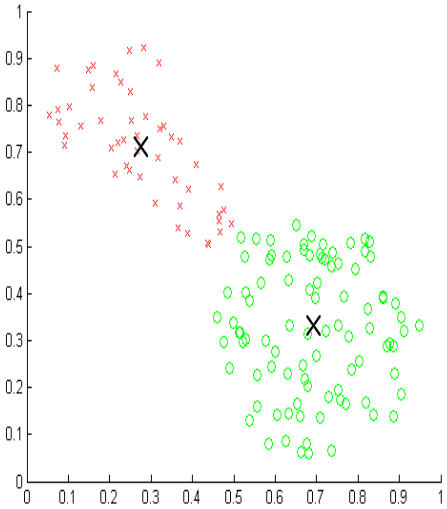

Figure-2

IV.RESULT

The result shown in figure-1 and figure-2 is quite good as compare to FCM and PCM. The PFCM algorithm overcome the problem of Overlapping of the coincident clusters.

V.CONCLUSION

We have implemented the PFCM in Matlab taking all the considerations in mind that about the membership value and typicality.

REFERENCES

[1] Nikhil R. Pal,Kuhu Pal, James M. Keller, and James C. Bezdek," A Possibilistic Fuzzy c-Means Clustering Algorithm,"*IEEE Trans. on Fuzzy Systems*, vol. 13, no. 4, pp. 517-530, Aug. 2005.

[2] Jian Cui, Qiang Li, Jun Wang and Da-Wei Zong," Research on Selection Method of the Optimal Weighting Exponent and ClusteringNumber in Fuzzy C-Means Algorithm ," *International Conference on Intelligent Computation Technology and Automation*, pp. 104-107, 2010.

[3] Jiang-She Zhang and Yiu-Wing Leung," Improved Possibilistic C-Means ClusteringAlgorithms," *IEEE Trans. on Fuzzy Systems*, vol. 12, no. 2, pp. 209-217, April 2004.

[4] S. Nefti and M. Oussalah," Probabilistic-Fuzzy Clustering Algorithm," *IEEE lntemational Conference on Systems, Man and Cybemetics*, pp. 4786-4791, 2004.

[5] Jiabin Deng, JuanLi Hu, Hehua Chi and Juebo Wu," An Improved Fuzzy Clustering Method for Text Mining," *Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, pp. 65-69, 2010.

[6] Maryam hajiee," A New Distributed Clustering Algorithm Based on K-means Algorithm," *3rd International Conference on Advanced Computer Theory and Engineering*, pp. 408-411, 2010.

[7] Jiang Haiyan and DU Min," Research on the Detection of Gold Immunochromatographic Assay by the Image Histogram Feature Vectors and Fuzzy C-means,"*EightInternational Conference on Fuzzy Systems and Knowledge Discovery*, pp. 467-471, 2011.

[8] JialunLin ,Xiaoling Li and Yuan Jiao," Text Categorization Research Based on Cluster Idea," *Second International Workshop on Education Technology and Computer Science*, pp.483-486, 2010.