THESIS APPROVAL

GRADUATE SCHOOL, KASETSART UNIVERSITY

Doctor of Philosophy (Computer Science)

DEGREE

Computer Science — **FIELD**

Computer Science — **DEPARTMENT**

TITLE: Automatic Exponential Fuzzy Clustering with Outlier Detection

NAME: Mr. Kiatichai Treerattanapitak

THIS THESIS HAS BEEN ACCEPTED BY

_____ **THESIS ADVISOR**

(Associate Professor Chuleerat Jaruskulchai, Ph.D.)

_____ **DEPARTMENT HEAD**

(Assistant Professor Sirikorn Channual, M.S.)

APPROVED BY THE GRADUATE SCHOOL ON _____

_____ **DEAN**

(Associate Professor Gunjana Theeragool, D.Agr.)

THESIS

AUTOMATIC EXPONENTIAL FUZZY CLUSTERING WITH OUTLIER DETECTION

KIATICHAI TREERATTANAPITAK

A Thesis Submitted in Partial Fulfillment of
the Requirements for the Degree of
Doctor of Philosophy (Computer Science)
Graduate School, Kasetsart University

2012

Kiatichai Treerattanapitak 2012: Automatic Exponential Fuzzy Clustering with Outlier Detection. Doctor of Philosophy (Computer Science), Major Field: Computer Science, Department of Computer Science. Thesis Advisor: Associate Professor Chuleerat Jaruskulchai, Ph.D. 111 pages.

The degree of membership is the key element to achieve high quality of Fuzzy Clustering algorithm. Traditional algorithm like Fuzzy C-Means (FCM) does not produce the degree of membership to reflect the actual level of belonging in some situations. In addition, performing clustering on the dataset containing noise and outliers leads to inaccurate clustering result due to cluster centroids are influenced and shift away from their actual positions and sometimes generates coincident clusters. Furthermore, parameters setting is difficult for inexperience users to operate the clustering algorithm. In this thesis, Exponential Fuzzy Clustering (XFCM) is proposed based on the three types of degree of membership concept to improve its representation. Additionally, the problem of noise and outlier are handle by combining the Possibilistic approach with Exponential Fuzzy Clustering. To solve the problem of setting number of clusters and estimating fuzzifier, the Agglomerative Fuzzy Clustering (AFC) is proposed with a single parameter. Various experiments were setup to validate XFCM, PXFCM and AFC performance. The experiments for XFCM were carried out to measure the prediction of errors by Mean Absolute Error (MAE) on Collaborative Filtering. The results showed that XFCM outperforms FCM by 5.2-9.8%, FCME by 1.0-6.1%, the Item-based method by 2.7-6.9% and SVD by 1.0-3.0% for 100K and 1M MovieLens dataset. PXFCM produced minimum centroid errors comparing to other algorithms and did not generate coincidence clusters. In the outlier detection perspective, the XOF that calculated based on the residual distance yielded the better result than other outlier detection algorithms. AFC also selected the right value of fuzzifier and number of cluster parameters for fuzzy clustering. This method can be used to automate the algorithm and it is easy to operate by novice.

_____ _____ ___/___/___

Student's signature Thesis Advisor's signature

ACKNOWLEDGEMENT

I would like to grateful thank and deeply indebted to Associate Professor Dr. Chuleerat Jaruskulchai my thesis advisor for advice, encouragement and valuable suggestion for completely writing of thesis. I would sincerely like to thank Professor Dr. Chidchanok Lursinsap and Assistant Professor Dr. Arnon Rungsawang my committees. I also would like to thank Associate Professor Dr. Zengyou He for kindly sharing the dataset for using in this thesis. I am heartfelt thanks to every teachers and officers in the Department of Computer Science.

I am especially appreciated my parents, my sister and brothers for their continuing encouragements. Finally, I am deeply appreciated to Mrs. Sarika Treerattanapitak who always devotes time and supports and also my daughters who always gives me heartfelt love during my graduate study.

Kiatichai Treerattanapitak

September 2012

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

Figure **Page**

LIST OF ABBREVIATIONS

k	=	number of clusters
N	=	total number of data point in the dataset
ε	=	termination coefficient
x_i	=	the i^{th} data point
v_j^0	=	the initial cluster's center or centroid
v_j	=	cluster's center or centroid vector
M	=	length of the i^{th} data point
d_{ij}, d_{iu}, $\left\|x_i - v_j\right\|$	=	distance function or similarity
N_j	=	the total number of members in cluster j
J_X	=	the objective function
m, η	=	the fuzzifier parameter or weight
μ_{ij}	=	the membership function or degree of membership
w	=	weight factor
$\delta(v_j,v_k)$	=	centroid similarity
$P_{u,i}$	=	the prediction rating of user u on item i
$S_{i,N}$	=	similarity measure of all similar items N and item i
$R_{u,N}$	=	given rating score by user u to items N.
$sim(i,j)$	=	similarity of each item i to cluster j.
$R_{u,i}$, $R_{u,j}$	=	rating score by user u to item i and j respectively
$\overline{R_u}$	=	average rating of user u.
δ	=	standard deviation

AUTOMATIC EXPONENTIAL FUZZY CLUSTERING WITH OUTLIER DETECTION

INTRODUCTION

Clustering technique has been proven to be a very efficient unsupervised learning tool to analyze unstructured data in classification problems. Clustering has its objective to separate unlabeled data into finite and discrete sets. Data in the same cluster are more similar than data in the other clusters.

Basically, Partitioned Clustering, K-Means (MacQueen, 1967) or Crisp (Miyamoto *et al*., 2008) or Hard C-Mean (Oliveira and Pedrycz, 2007) is the most popular method using in many scientific fields because of ease to implement, efficiency, simplicity and empirical success in implementation. Even this method has been developed more than 40 years. There are still having rooms to improve the cluster efficiency (Jain, 2010). In this thesis, the word "K-Means" will be used for crisp clustering and the word "C-Means" will be used for fuzzy clustering. K-Means assigns data exactly to one cluster, thus data with overlapped clusters do not well partition by K-Means. The fuzzy version of K-Means (Fuzzy C-Means or FCM) is then proposed by incorporating data to be member of all clusters but in different degree of membership (Dunn, 1973; Bezdek, 1981). By processing FCM, number of clusters, selection of initial centroids and fuzzifier or fuzzy exponent are required as the input. The difficulty to estimate these parameters prevents FCM to produce a good result especially the dataset where useful information to determine parameters is unavailable. In addition, the distribution of degree of membership produced by FCM algorithm does not well represent the degree of belonging. This leads unrelated data to be assigned to the clusters.

Datasets that contain noises and outliers are even more difficult to clustering because centroids are influenced by these abnormal data points and shifting away from

their true positions. With this reason, FCM is very sensitive to noise. In some situation whereby quality of clustering is crucial such as clustering problem in Collaborative Filtering (CF) domain and recommendation problem, they require only truly related data points in order to produce good recommendations. One possible solution to resist noise and outlier is to use the actual data points as the centroid as K-Medroid algorithm. Nevertheless, this method leads to a lot of computation to find the actual center (Mei and Chen, 2010).

FCM allocates unrelated data to the clusters by assigning lower degree of membership. However, it does not enough to make high quality for recommendation system. There are several researches to improve this weakness but most of them derive from FCM and always come with additional parameters which make the algorithm more complicate to use (Pal *et al*., 1997; Pal *et al*., 2005). FCM and some of its variants are not the right algorithm to deal with this problem because the probability condition that sums the degrees of membership of data to all clusters is one. If dataset contains a lot of noise and outliers, these data points are treated as ordinary data points by forcing to be member of a cluster.

Parameters setting can cause clustering to produce incorrect result. Poor seeds selection leads FCM gets struck in local minima. (Arthur and Vassilvitskii, 2007) Initialization that includes noises and outliers, causes the clusters to be formulated by abnormal data points instead of useful data points. In addition, incorrect number of clusters does not reflect the structure of dataset. The most widely used methods to find the number of clusters is to validate clustering result with Cluster Validation Index (CVI) (Xie and Beni, 1991) These methods are computationally expensive with numerous of validation that perform on every clustering results and obtaining the number of clusters based on the best index. These methods do not practical for automatic clustering which is a subject of this thesis. Furthermore, these methods do not applicable to validate the value of fuzzifier. In general the value of fuzzifier is in range between [1.5,2.5] (Wu, 2012) or setting to 2 in most cases (Pal and Bezdek, 1995; Zhang *et al*., 2008). The actual value for particular dataset can be estimated by a relation of number of data and their dimensions (Schwammle *et al*., 2010).

Thus, this thesis aims to develop a partitioned clustering algorithm that immunes to noise and outlier and requires minimal input of parameters by automatically find appropriate parameters.

Contribution of this thesis

Clustering is a useful algorithm in classification problem. It is unsupervised learning that means it is capable to separate data without any knowledge to the dataset. However initial parameters are required and could not be estimated by novice persons. As an objective of this thesis aims to develop an automate clustering procedure thus anyone is able to perform the clustering. In addition, the outliers in the dataset are also detected and properly manipulated in order to perform further analysis on those detected outliers.

During the thesis progression, several components of this thesis have been published to the international conferences and journals as follows. Firstly, Clustering was studied on Collaborative Filtering domain. By using this method, it reduces time complexity for the prediction in Collaborative Filtering. This idea was published in following article.

Treerattanapitak, K and C. Jaruskulchai. 2009. Items Based Fuzzy C-Mean Clustering for Collaborative Filtering. J. Information Tech. 10 : 30-34.

Secondly, the first article was extended to improve quality of prediction by using Entropy based Fuzzy C-Mean. This idea was published in following article.

Treerattanapitak, K and C. Jaruskulchai. 2009. Entropy based Fuzzy C-Mean for Item-based Collaborative Filtering. In 9th International Symposium on Communication and Information Technology. pp. 881-886.

Thirdly, new clustering algorithm was developed based on Exponential function in order to improve prediction quality. This idea was published in following article.

Treerattanapitak, K. and C. Jaruskulchai. 2010. Membership enhancement with exponential fuzzy clustering for collaborative filtering, In Proc. of the 17th Intl. conf. on Neural info. Processing, Springer-Verlag, Berlin, Heidelberg, pp. 559-566.

Fourthly, the new algorithm was technically studied with extensive experiments and published in the international journal as follows.

Treerattanapitak, K. and C. Jaruskulchai. 2012. Exponential Fuzzy C-Means for Collaborative Filtering, J. Comp Sci and Tech. 27, No.3, pp. 567-576.

Furthermore, this algorithm was extended by Possibilistic approach to become Possibilistic Exponential Fuzzy Clustering and published in following article.

Treerattanapitak, K. and C. Jaruskulchai. 2011. Outlier detection with Possibilistic Exponential Fuzzy Clustering, 8th Int. Conf. on Fuzzy Sys. and Know. Discovery, pp.453-457.

Possibilistic Exponential Fuzzy Clustering was technically studied with extensive experiments and being published to Journal of Computer Science and Technology (JCST). Finally, to automate the clustering, number of clusters need to be automatically obtained. Thus Agglomerative FCM was studied and generalized to use with FCM variants algorithm. The article for this idea has been accepted by ICONIP 2012 and to be published in Lecture Notes in Computer Science.

OBJECTIVES

This study has following objectives:

1. To develop new partitioned fuzzy clustering algorithm by modifying the objective function based on Exponential function.

2. To improve Exponential fuzzy clustering from first objective by enabling outlier detection capability.

3. To develop an automation procedure for the fuzzy clustering algorithm.

Thesis Outline

This Thesis organizes as follows. Firstly, related works were reviewed in Literature Review section which described the fuzzy clustering and its variants, addressing noise and outliers in the relation of fuzzy clustering as well as reviewed related works of number of clusters, fuzzifier and initialize seed parameters for fuzzy clustering. Secondly, this thesis proposed the new algorithm that developed based on Exponential function. This new algorithm was discussed on its properties and addressed a problem for improvement. In order to overcome the limitation that sum degree of membership to one, this thesis integrated Possibilistic approach to proposed algorithm. In addition, this thesis developed a new procedure in order to automate fuzzy clustering. Thirdly, the experiments were performed on both new algorithms and validated the automate procedure with comprehensive experiments. Finally, the conclusion and recommendation were provided at the end of this thesis.

LITERATURE REVIEW

Generally, Clustering is broadly classified into two categories; Hierarchical Clustering build the nested cluster structure either by divisive or agglomerative of similar data. There are four basics Hierarchical Clustering algorithms i.e. Single link merges two clusters by similarity from most similar (Sneath, 1957; Sneath and Sokal, 1973); Complete link merges two clusters by similarity from most dissimilar (Sorensen, 1948; King, 1967); Average link merges two clusters by average similarity from both clusters (D'andrade, 1978); Centroid link merges clusters by similarity from all members (Ward, 1963).

Partitioned Clustering (MacQueen, 1967) algorithm performs in iteration by assignment data to the closet cluster and computes cluster center or centroid according to their membership. Algorithm produces optimum solution by minimizing the objective function that mathematically measures in term of overall distance between each data point and each cluster's centroid. Each clustering has its own advantages and disadvantages as follows.

Table 1 Advantages and disadvantages between Hierarchical and Partitioned Clustering

Hierarchical Clustering	Partitioned Clustering
• Similarity calculates in entire dataset. • Does not require initial number of clusters for the input but interpretation for the number of clusters in final output is required. • Process in single iteration. • Time complexity is $O(N^2)$	• Similarity calculates with centroids. • Require number of clusters as input. • Process in several iterations to get optimum result. • Time complexity is $O(Nkl)$

The most popular clustering is probably K-Means which is one of the partitioned clustering (Jain, 2010). K-Means allocates each data point into an exactly one cluster. The algorithm begins with empty clusters and initial seeds. These seeds are selected from data by randomization or other equivalent technique and assign to clusters as initial members. Other data points then allocate to their closet clusters according to the similarity. At the end of iteration, each cluster computes the centroids regarding to their membership. These processes are repeated until the objective function does not changed or there is no interchange of members between clusters.

For dataset with overlapped clusters, the data points locate within the overlapped area could be assigned to multiple clusters. Performing K-Means on these kinds of dataset leads to disadvantage that data points are assigned exactly to one cluster. Fuzzy Clustering derives from K-Means to overcome this issue by incorporating the degree of belonging to each data points. In fact, K-Means is a type of fuzzy clustering whereby degree of membership is 1 for the nearest cluster and is 0 for other clusters. Fuzzy Clustering is described as follows:

1. Variants of Fuzzy Clustering

The Fuzzy approach involves uncertainty i.e. the data could not belong to one cluster nevertheless all data belong to all clusters with the different degree of membership. The most classical fuzzy version is FCM. The FCM's procedure to partition data is illustrated in Figure 1.

1.1 Fuzzy C-Means (FCM)

In contrast to K-Mean (MacQueen, 1967), FCM (Dunn, 1973; Bezdek, 1981) incorporates the degree of belonging in term of the degree of membership (μ_{ij}) which its value between [0,1]. FCM has objective function derivative from K-Means by introducing fuzzifier *m* as(1).

$$J_{FCM} = \sum_{j=1}^{k}\sum_{i=1}^{N} \mu_{ij}{}^{m} d_{ij}{}^{2} \text{ ; where } m \in (1,\infty), \sum_{j=1}^{k} \mu_{ij} = 1 \quad (1)$$

STEP1 Predefined parameters: This step is to define the required parameters to process in the algorithm. They are; number of clusters (k), fuzzifier parameter (m), termination coefficient (ε) for a termination process and initialize k centroids (v_j^0) for each cluster j.
STEP2 Allocation the data: This step is to assign data into their related clusters at different degree of membership according to the similarity.
STEP3 Update Centroids: Once all data are allocated, the centroids get update regarding to their members.
STEP4 Validate the Termination Criteria: Repeat STEP2 until the desire condition is met then algorithm is terminated.

Figure 1 Procedure of fuzzy clustering.

In order to minimize objective function, Lagrange Multiplier is used to find solution for (1) μ_{ij} and v_i can be computed accordingly from (2) and (3) (See Appendix A for more details).

$$\mu_{ij} = \frac{1}{\sum_{u=1}^{k} \left(\frac{d_{ij}{}^{2}}{d_{iu}{}^{2}} \right)^{\frac{1}{m-1}}} \quad (2)$$

$$v_j = \frac{\sum_{i=1}^{N} (\mu_{ij}{}^{m} x_i)}{\sum_{i=1}^{N} \mu_{ij}{}^{m}} ; 1 \le {}_j \le k \quad (3)$$

If fuzzifier setting is closer to one, the objective function in (1) will rely only to the distance which is the same objective function of K-Means as (4). FCM algorithm process in similar manner to K-Means. The algorithm terminates by

validation of the change of objective function at iteration *t* and previous iteration *t-1*.If J_{FCM}^{t}-$J_{FCM}^{t-1}<\varepsilon$ where ε is a small value, the algorithm is terminated.

$$J_{K-Means} = \sum_{j=1}^{k}\sum_{i=1}^{N} d_{ij}^{\ 2} \tag{4}$$

There are other fuzzy variations that develop for specific purposes especially to improve the clustering quality. These algorithms redefine the objective function with different mathematical formation as described below.

1.2 Fuzzy C-Means with Entropy Regularization (FCME)

This type of FCM combines an Entropy term that measures uncertainty of information content in the Information Theory. Miyamoto *et al.* (1997) proposed addition term with Entropy Regularization to the objective function in (1) and entropy type of criterion has considered by Miyamoto *et al.* (2004). In order to improve consistency clustering result of FCM, the objective function is modified as (5)

$$J_{FCME} = \sum_{j=1}^{k}\sum_{i=1}^{N} \mu_{ij} d_{ij}^{\ 2} + m\sum_{j=1}^{k}\sum_{i=1}^{N} \mu_{ij} \log \mu_{ij}\,; \text{where } m>0, \sum_{j=1}^{k} \mu_{ij} = 1 \tag{5}$$

By minimizing the objective function, the optimum solution that obtains by Lagrange Multiplier for μ_{ij} and v_i can be computed accordingly from (6) and (7).

$$\mu_{ij} = \frac{\exp(-m \cdot d_{ij}^{\ 2})}{\sum_{u=1}^{k}\left(\exp(-m \cdot d_{iu}^{\ 2})\right)} \tag{6}$$

$$v_j = \frac{\sum_{i=1}^{N}(\mu_{ij} x_i)}{\sum_{i=1}^{N}\mu_{ij}}; 1 \le j \le k \qquad (7)$$

The algorithm tries to minimize the objective function by minimizing within cluster dispersion in the first term and maximizing the negative entropy that has its important role to the contribution of data to cluster. The *m* parameter is used to control the effect of entropy term. When *m* is very large, the algorithm assigns data to a single cluster which behaves similar to K-Means.

1.3 Possibilistic Fuzzy C-Means (PFCM)

As aforementioned, FCM is sensitive to noise (See Figure 3). In addition, sum of the degrees of membership across all clusters for each data point to one is a limitation that turns the abnormal points to be a member of a cluster. In order to overcome this situation, possibilistic approach is integrated to FCM to relax this condition (Krishnapuram, 1993). The objective function for Possibilistic Fuzzy C-Means (PFCM) can be formulated as (8)

$$J_{PFCM} = \sum_{j=1}^{k}\sum_{i=1}^{N}\mu_{ij}{}^{m} d_{ij}{}^{2} + \sum_{j=1}^{k}\lambda_j \left(\sum_{i=1}^{N} 1 - \mu_{ij}\right); \text{where } \lambda_j > 0, \sum_{j=1}^{N}\mu_{ij} < N \qquad (8)$$

The optimum solution of membership degree and centroid are derived the same way by minimizing the objective function as other Fuzzy Clustering. The optimum membership degree and centroid is in (9) and (10) respectively.

$$\mu_{ij} = \frac{1}{1 + \left(\frac{d_{ij}{}^{2}}{\lambda_j}\right)}, \lambda_j = K \frac{\sum_{i=1}^{N}(\mu_{ij}{}^{m} d_{ij}{}^{2})}{\sum_{i=1}^{N}\mu_{ij}{}^{m}} \qquad (9)$$

$$v_j = \frac{\sum_{i=1}^{N} (\mu_{ij}{}^m x_i)}{\sum_{i=1}^{N} \mu_{ij}{}^m}; 1 \le {}_j \le k \qquad (10)$$

λ_j is a positive number and K is an adjustable weight and typically be one. Algorithm does not force data to be membered of any cluster. This method is useful when dataset containing some abnormal points since these data points will be assigned with lower degree of membership. Even though the algorithm sounds very promising but the objective function (8) is truly minimized if all clusters are identical (coincident clusters). The reason is that the degree of membership in (9) depends only on the distance between data and particular cluster without considering the distance to other clusters (Barni *et al*., 1996).

Coincident cluster is a circumstance that some centroids are propagated their location during the clustering execution to share the same location with other centroids. This can be a problem especially small datasets because all data points can be lumped into one cluster. However, coincident cluster can be advantageous in some situation. For example, when clustering starts with a large value of number of clusters and final result contains some coincident clusters. This may indicate the correct number of clusters (Pal *et al*., 2005).

1.4 Unsupervised Possibilistic Fuzzy C-Means (UPC)

Unsupervised Possibilistic Fuzzy C-Means (UPC) is another Possibilistic approach based on the partition coefficient and partition entropy (Yang and Wu, 2006). The objective function is defined as (11)

$$J_{UPC} = \sum_{j=1}^{k}\sum_{i=1}^{N} \mu_{ij} d_{ij}{}^2 + \frac{\beta}{m^2\sqrt{k}} \sum_{j=1}^{k}\sum_{i=1}^{N} \mu_{ij} \log \mu_{ij} - \mu_{ij}; \sum_{j=1}^{N} \mu_{ij} < N \qquad (11)$$

The optimum solution of membership degree and centroid are derived in the same way as other Fuzzy Clustering by minimizing the objective function using Lagrange Multiplier. The optimum membership degree and centroid are in (12) and (13) respectively.

$$\mu_{ij} = \exp\left(-\frac{m\sqrt{k}d_{ij}^{\ 2}}{\beta}\right), \beta = \frac{\sum_{i=1}^{N}\left|x_i - \bar{x}\right|^2}{N}, \bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} \quad (12)$$

$$v_j = \frac{\sum_{i=1}^{N}(\mu_{ij}^{\ m} x_i)}{\sum_{i=1}^{N}\mu_{ij}^{\ m}}; 1 \le {}_j \le k \quad (13)$$

Although this method minimizes the impact of noisy data points by assigning very low membership degree, these data points exist in the output without separation from other good data points and eventually influence the centroids as described in Figure 3.

1.5 Analysis of fuzzy clustering

Each clustering algorithm described above has its own advantage. FCM is an improved version of K-Means whereby data belong to all clusters with the different degree of membership. FCM and its variants work well especially in overlapped cluster environment. Data points locate in overlapped area belong to all overlapped clusters. Allocation of data in this area exactly to a cluster does not sound correct. Other variants are the improved partition quality of FCM. For example, a dataset with one dimension has three clusters with centroid 0, 1 and 2. This dataset is performed clustering with fuzzifier value sets to 2 for all algorithms. The degree of membership for cluster with centroid 0 is produced by each clustering as illustrated in Figure 2.

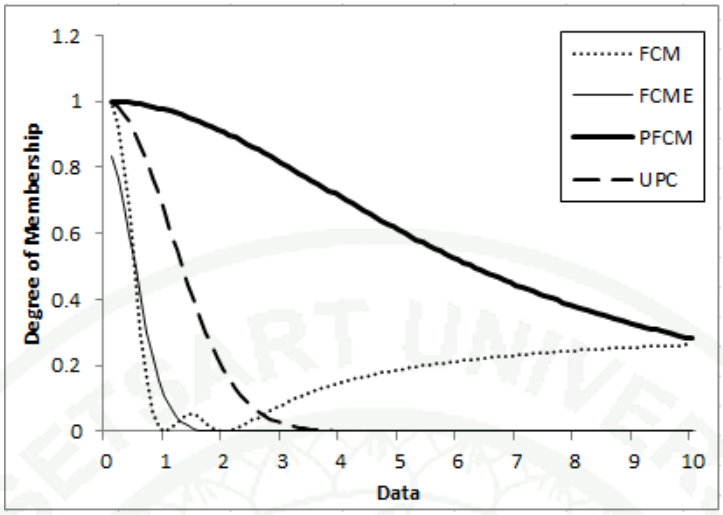

Figure 2 Degree of membership produced from FCM for 1 dimension data.

From Figure 2, The X-Axis represents the one dimension data or similarity to the cluster and the Y-Axis is the degree of membership. FCM assigns zero degree of membership to data 2 and assigns higher degree of membership to the data locates further which is not make sense. In fact, it should present monotonic tendency with fuzzifier. By another mean, the distribution of degree of membership of FCM is higher when compares to other algorithms since higher degree of membership could be assigned to dissimilar data points.

FCME, PFCM and UPC add the second term on top of FCM's objective function. These terms are mostly to improve partition quality by decreasing the distribution of the degree of membership. FCME uses the Entropy term to improve the assignment of the degree of membership with monotonic function thus higher dissimilar data will not assign to the cluster (Data above 2 get very low degree of membership) but the closet data point does not fully assigned to the cluster (Data 0 is assigned to the cluster with 0.8 degree of membership). PFCM breaks the sum to one condition. Adding the second term to objective function leads PFCM to present the monotonic tendency with degree of membership. The level of degree of membership is slightly decreased when dissimilarity is increased. Nevertheless, all data points can be assigned to the cluster (Data 10 is assigned to the cluster with 0.3 degree of membership). The second term of UPC behaves similar to FCME by not assigning

dissimilar data to the cluster and the closet data point is assigned to cluster with reasonable degree of membership. Since Data 1 and Data 2 are the centroids of the other two clusters, they are assigned to this cluster instead of fully belong to those cluster. All of the additional terms adding to the objective function have different advantages and disadvantages that can be summarized in Table 2.

Table 2 Advantages and disadvantages of addition terms for Fuzzy Clustering Variants

Algorithm	Term	Advantage	Disadvantage
FCME	$\sum_{j=1}^{k}\sum_{i=1}^{N}\mu_{ij}\log\mu_{ij}$	Improve partition quality	Closet data point not fully belongs to cluster
PFCM	$\sum_{j=1}^{k}\lambda_j(\sum_{i=1}^{N}1-\mu_{ij})$	Improve partition quality	Allocation of Unrelated data
UPC	$\frac{\beta}{m^2\sqrt{k}}\sum_{j=1}^{k}\sum_{i=1}^{N}\mu_{ij}\log\mu_{ij}-\mu_{ij}$	Improve partition quality	Centroid not fully belong to cluster

2. Noise and Outlier

Clustering dataset with noise and outliers leads to inaccurate result because they are influence centroids when assign to clusters. Thus, noise and outliers must be properly handled for clustering. In this thesis noise and outlier are defined as follow (Chandora *et al*., 2009) .

"Noise can be defined as a phenomenon in data that is not of interest to the analyst, but acts as a hindrance to data analysis."

"Outlier is an unobserved data that numerically deviates from the rest of data. Outliers look like ordinary data but appear to be inconsistent with others in the dataset"

The difference between noise and outlier is that outliers are interested for finding information inside them while noise is uninteresting data. The important of the outlier detection is the fact that outliers hold uncovered information and sometime may contain the new knowledge for application. For examples, traffic patterns to access the network could be from a hacked computer or unauthorized sources (Kumar, 2005), patterns of using credit card could indicate transaction with stolen credit cards (Aleskerov *et al.*, 1997).

Clustering the dataset with noise and outlier causes the centroid to shift position. For example a 2-dimension dataset consists of 16 points in 2 clusters as illustration in Figure 3. Both clusters have centroids in A1 and B1. When this dataset has a group of outliers locate around (10,10). The centroids are shifted to A2 and B2 accordingly. This is because FCM attempts to allocate them to the clusters and resulting to the centroid shift.

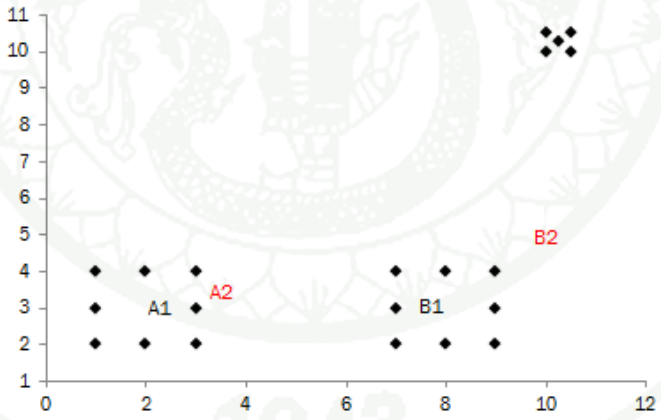

Figure 3 Impact of outliers for 2 dimension data.

In order to handle noise and outliers, these abnormal points must be properly handled prior to clustering. This is the subject of Outlier Detection which categorized into six categories by their approaches as follow.

2.1 Statistical approach

This approach attempts to detect the noise by modeled the dataset with statistical technique. The outlier data are usually deviated very far away from the mean. The simplest technique uses the standard deviation (δ) to declare the outliers whereby the data instances that locate more than 3δ are the outliers (Shewhart, 1931). Some applications demonstrate the usage of statistics method to detect, isolate and remove outlier and noisy data on whether dataset (Shahi, et al., 2009). These methods can operate in unsupervised manner and yield good result with the degree of confident. Nevertheless, these methods may not practical for some dataset because many datasets do not fit in one particular model thus they do not work well in large dataset. (Hodge and Austin, 2004)

2.2 Clustering approach

This technique groups similar data into the same cluster and isolates outliers from normal data points such as determining small cluster as outlier (Al-zoubil, 2010) or robust clustering by separating the noisy data into the noise clusters (Dave, 1991) Another approach performs posterior clustering procedure based on entropy membership to filter out the outliers (Li, *et al*., 2009). Some clustering algorithms do not force outliers to belong to any clusters and detect unassigned data as outliers such as DBSCAN (Ester, *et al*., 1996). These algorithms operate in unsupervised mode thus additional knowledge of the dataset is not required. In addition, clustering algorithms do not search similar data by computing similarity in entire dataset but between data and centroids. Clustering methods are not computational expensive. Nevertheless, clustering is optimized to capture the cluster structure more than to detect outliers. Thus, the clustering results may not optimize for outlier detection. Clustering may ineffective especially when outliers form significant clusters among themselves.

2.3 Nearest Neighbor approach

Nearest Neighbor approach associates each data point with outlier scores that computed by examining the main characteristics of data in a group. A data point with higher outlier score usually deviates from most data and is considered to be outliers. These methods broadly categorize into two groups. First group is k^{th} Nearest Neighbor that defines an outlier score based on its distance to its k^{th} nearest neighbor and select *n* instances with largest outlier scores as outliers (Ramaswamy, et al., 2000; Knorr, et al., 1998). Second group determines outliers by relative density of data points. The data points are considered as outliers if they lay in low density of neighborhood. The outlier score for each data point is computed based on number of surrounding points and defines as Local Outlier Factor (LOF). LOF is defined by (14) and (15)

$$lrd_{MinPts(p)} = \frac{\left|N_{MinPts(p)}\right|}{\sum_{o \in N_{MinPts(p)}} Sim_{MinPts(p,o)}} \tag{14}$$

$$LOF = \frac{\sum_{o \in N_{MinPts(p)}} \frac{lrd_{MinPts(o)}}{lrd_{MinPts(p)}}}{\left|N_{MinPts(p)}\right|} \tag{15}$$

Where $Sim_{MinPts(p,\,o)}$ is the similarity between data point *p* and its neighbor point *o*, $N_{MinPts(p)}$ is the total number of neighbor points of *p*. Normal data will have LOF approximately equal to 1 and outliers are otherwise (Escalante, 2005; Kaur ,2008; Breunig *et al.,* 2000) . LOF requires minimum number of points to determine the neighbor. Nevertheless, this method is computational expensive from a large number of queries that attempts to find the neighbors for each data point. In addition, the algorithm may fail if the input parameter for minimum points is not properly defined.

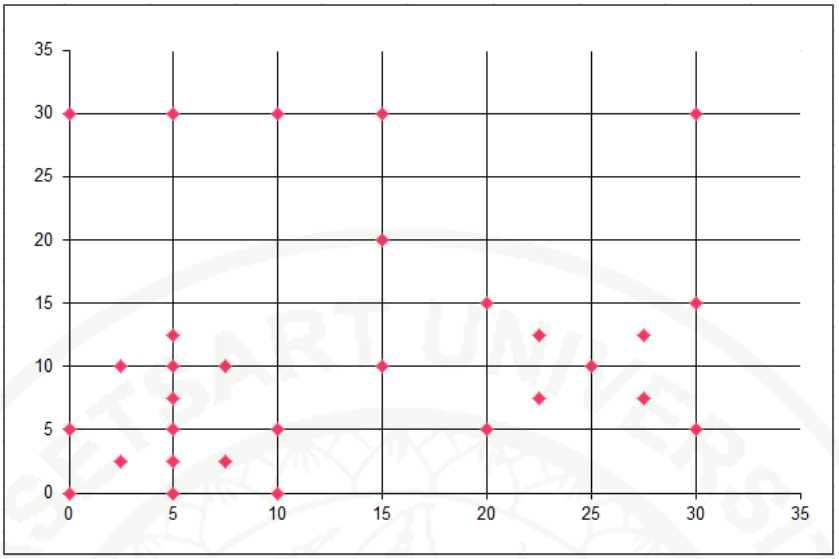

Figure 4 Sample dataset for LOF calculation.

An example of calculation LOF can be demonstrated as follow. Assume there is a dataset as Figure 4. The task is to compute LOF for (10,30) when the minimum points set to 5.

Firstly, similarity between each pair of data is calculated. In this case all other data are computed similarity against (10,30). Secondly, all similarity are sorted to obtain the most 5 similar data. The computation returns (5,30), (15,30), (0,30), (15,20) and (20,15) are the most similar data with similarity value of 5,5,10,11.18 and 18.03 respectively. Thirdly, compute *lrd* by (14) which is $\frac{5}{5+5+10+11.18+18.03} = 0.1016$. Fourthly, compute *lrd* for the 5 neighbors which yield 0.097, 0.091, 0.075, 0.102 and 0.153 respectively. Lastly, compute LOF for the data by (15), which is $\frac{0.1016}{(0.097+0.091+0.075+0.102+0.153)/5} = 1.021$

Nearest Neighbor methods are unsupervised and easy to implement but these methods rely on distance measure and computation expensive from excessive similarity calculation of every pairs of data.

2.4 Classification approach

Classification approach commonly consists of two phases. The first phase (Training) attempts to build a model by learning from dataset that already labeled as the outliers. The second phase (Classification) is to separate outliers out of normal data points by the learning from training model. Classification methods can be integrated with the powerful data mining algorithm to distinguish the difference of outliers. The classification phase is usually fast because these methods test the data against pre-built model. However these methods rely on quality of label which is not possible in some situation especially when information of dataset is unavailable. Some recent researches on classification based approaches such as using Rule based (Angiulli *et al.*, 2008).

2.5 Information Theoretic approach

The assumption of this approach is that the outliers containing in dataset influence content of dataset to have higher complexity. Removing outliers decreases the level of complexity of the dataset. The goal of this approach is to find removal subset that minimizes complexity. Complexity can be measured by Information Theory techniques such as Kolomogorov complexity (Li and Vitanyi, 1993). These methods operate in unsupervised mode but these methods rely on the measuring of complexity and also difficult to associate outlier score.

2.6 Spectral approach

This approach assumes that data and outliers are separable by approximation into lower dimensions. Generally, this approach is to find subspace where outliers can be identified (Agovic, *et al.*, 2007 ; Bandyopadhyay *et al.*, 2008). Spectral methods perform in unsupervised mode and capable to handle data with high dimensions. Furthermore, these methods can be used as preprocessing to reduce dimensions for other outlier detection techniques. However, these methods are

computational expensive and effective in which outliers can be detected in lower dimensions.

3. Fuzzy Clustering Parameters

Outlier detection described in section 2 is usually a preliminary process to discriminate outliers from the normal data points prior to perform actual clustering. Clustering parameters such as fuzzifier parameters, number of clusters and initialization of clustering seed are crucial factors for effective clustering. As the objective of this thesis is to automate the clustering process, previous relevant researches are reviewed in these three subjects as follows:

3.1 Fuzzifier parameters

Basically, the role fuzzifier is to control the fuzziness of clustering algorithm. In case fuzzifier *m* is very high (FCM, UPC, PFCM) or *m* (FCME) is very low, each clustering yields the same result as K-Means and produces average degree of memberships if setting of *m* is opposite. In Figure 5 to Figure 8, the degrees of membership are produced by assuming that there are three clusters of one dimension data with centroid 0,1 and 2.

As aforementioned, FCM always allocates data to every cluster causing data points locate very far from clusters or irrelevant data are also assigned to a cluster as illustrated in Figure 5. Moreover, the noisy points are forced to be included in a cluster because FCM is associated with probabilistic condition. When *m* is increased, FCM produces the membership by *1/k*.

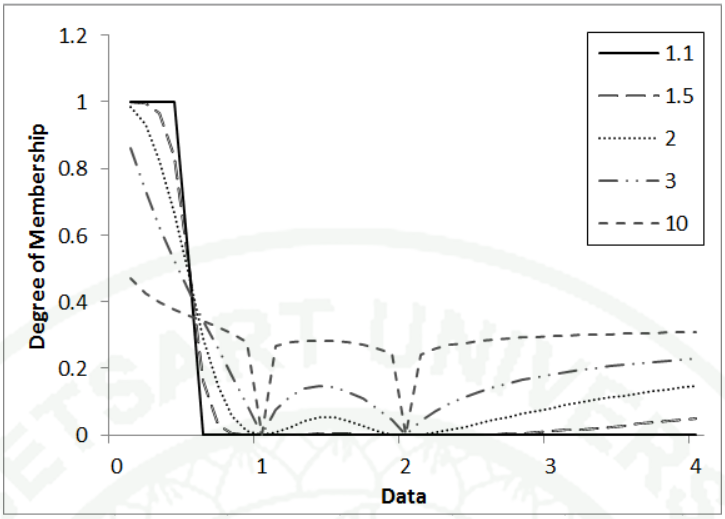

Figure 5 Degree of membership produced from FCM for 1 dimension data.

FCME uses different range of fuzzifier *m* and it is monotonically relation between *m* and degree of membership. If *m* is very low, the algorithm produces average the degree of membership by *1/k*. If *m* is very high, the result is the same as K-Means as illustrated in Figure 6. Noisy points will be included in the cluster if *m* is decreased. However, there are not many researches that study how *m* should be used.

For PFCM, this method does not have probabilistic condition thus PFCM produces membership degree without restriction. In general, PFCM ceases impact of noisy data by assigning these data points with lower value of membership degrees. However, these data points still belong to all clusters since the objective function is developed based on FCM. PFCM produces higher value of membership degrees for noisy data when *m* is increased. Membership degrees approach to 0.5 for all data when *m* is very high as illustrated in Figure 7.

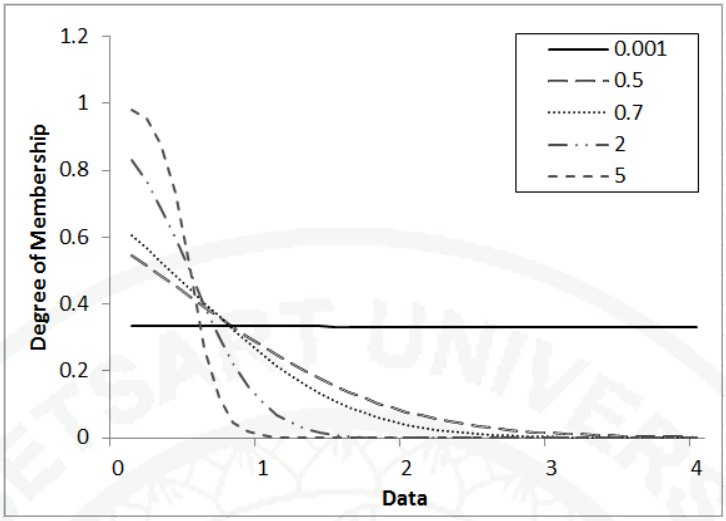

Figure 6 Degree of membership produced from FCME for 1 dimension data.

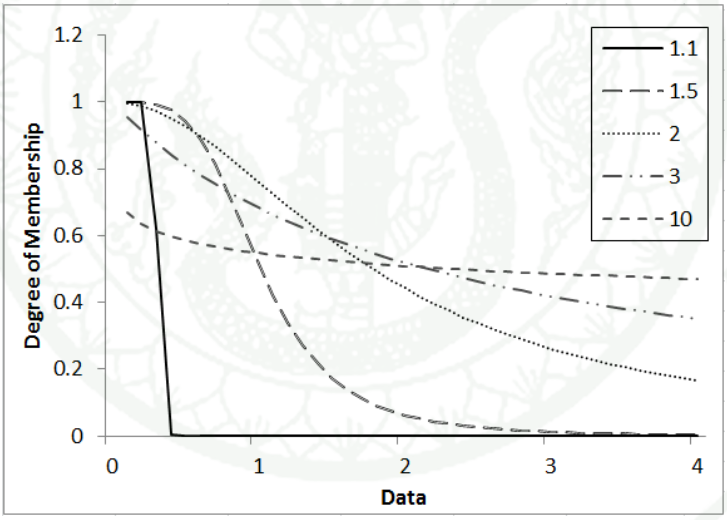

Figure 7 Degree of membership produced from PFCM for 1 dimension data.

UPC produces lower membership degree comparing to other methods. When *m* is increased, UPC filters almost all data out of clusters (*m* sets to 10) as illustrated in Figure 8. Although, this method is an improved version of PFCM but it still generates coincident clusters when the initial centroids are poor (Wu, *et al.*, 2010).

Impact of setting fuzzifier causes each clustering to produce result in different fuzziness level as summarized in Table 3.

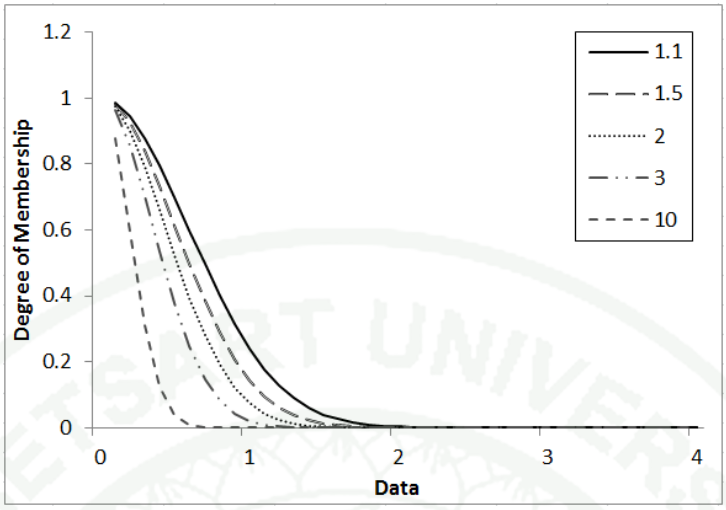

Figure 8 Degree of membership produced from UPC for 1 dimension data.

Table 3 Upper and Lower bound of degree of membership of each Fuzzy Clustering

Algorithm	Upper Bound	Lower Bound
FCM	$\lim_{m\to\infty} \mu_{ij} = 1/k$	$\lim_{m\to 1} \mu_{ij} = 1, d_{ij} = \min_{ij}\{d_{ij}\}$ $\lim_{m\to 1} \mu_{ij} = 0, d_{ij} > \min_{ij}\{d_{ij}\}$
FCME	$\lim_{m\to\infty} \mu_{ij} = 1, d_{ij} = \min_{ij}\{d_{ij}\}$ $\lim_{m\to\infty} \mu_{ij} = 0, d_{ij} > \min_{ij}\{d_{ij}\}$	$\lim_{m\to 0} \mu_{ij} = 1/k$
PFCM	$\lim_{m\to\infty} \mu_{ij} = 0.5$	$\lim_{m\to 1} \mu_{ij} = 1, d_{ij} = \min_{ij}\{d_{ij}\}$ $\lim_{m\to 1} \mu_{ij} = 0, d_{ij} > \min_{ij}\{d_{ij}\}$
UPC	$\lim_{m\to\infty} \mu_{ij} = 0$	$\lim_{m\to 1} \mu_{ij} = 1, d_{ij} = \min_{ij}\{d_{ij}\}$ $\lim_{m\to 1} \mu_{ij} = 0, d_{ij} > \min_{ij}\{d_{ij}\}$

There are two consequences in each clustering algorithm. If fuzziness level is low, all data tend to assign only to single cluster or algorithms produce crisp result like K-Means. On the other hand if fuzziness level is high, all data are assigned to all clusters at the same level. Algorithms produce the degree of membership for all data the same value or converge to a constant value.

In most cases, the fuzzifier set to two is a common usage in most application (Pal and Bezdek, 1995) but it generally works well only with FCM. For other FCM variants, these algorithms may have different level of fuzzifier. For example, fuzzifier value sets to 0.001 for FCME leads algorithm to produce average degree of membership.

In addition, some fuzzy clustering algorithms contain multiple weight parameters and those parameters do not thoroughly study in the articles (Pal, *et al*., 1997; Pal, *et al*., 2005; Wachs, *et al*., 2004). Hence those clustering algorithms required too much efforts to search by trial until a set of optimum parameters is found, for example, a fuzzy clustering variant algorithm is proposed but fine tuning parameters are not studied (Mei and Chen, 2010). The researches that study on the behavior of parameters could be divided into two groups as follows.

3.1.1 Incorporating with learning algorithms

This method incorporates with learning algorithm such as Neural Network (Borgelt and Kruse, 2003), Kohonen Network (Tsao and Bezdek, 1994) and Q-Learning (Oh, *et al.*, 2002). Fuzzifier parameter is adjusted by the learning process throughout the algorithm. Normally the adjusted equation has similar pattern as illustration in (16)

$$m = m_0 + w\Delta m \tag{16}$$

m_0 is an initialize fuzzifier. However, this method introduces new parameters w and m_0 which have to be specified before clustering process. This method is also called Fuzzy Learning Vector Quantization (FLVQ) which is widely used especially in image compression. Tsao and Bezdek (1994) proposed Fuzzy Kohonen Clustering Network (FKCN) by integrating Kohonen Network with FCM in order to fine tune the parameter during the clustering process or a robust method for

FKCN to resist the impact of noise in image segmentation (Lu, *et al.*, 2009). Another approach is to incorporate with learning algorithm to trade off parameters such as Q-Learning. The goal of Q-Learning tries to maximizing the reward. If parameters is adjusted in the right direction, it gets the reward. On the other hand, it gets the penalty if the adjustment is in the wrong direction (Oh, *et al.*, 2000) or incorporating with Neural network such as the method that learns parameters with Back Propagation technique (Borgelt and Kruse, 2003).

3.1.2 Adjustably by related information

Basically, the value of fuzzifier is in range between [1.5,2.5] and a suggestion $m=2$ is the preferred choice for FCM (Pal and Bezdek, 1995). However this value may not optimize for some dataset. Thus, some studies estimate fuzzifier parameters with mathematic reasoning such as theoretical studying the upper limit of m using Eigen value of the matrix (Yu *et al.*, 2004) or using number of dimensions and number of data to form an estimated function (Schwammle and Jensen, 2010). However these methods are applicable only FCM. Another method increases fuzzifier during the clustering execution by revising procedure with agglomerative method and obtains parameters by validating result with cluster validation (Li *et al.*, 2008). This method yields a very good result but requires massive executions in order to obtain the optimum fuzzifier. In addition, this method is applicable only to FCME.

3.2 Finding number of clusters

This problem is a fundamental question for clustering process. Wrong number of clusters leads to incorrect result and fails to interpretation. There are several methods to obtain the number of clusters that categorized by their approaches as follows.

3.2.1 Cluster Validation

Cluster validation technique is the most popular technique and it is an important step to validate the result in cluster analysis procedure (Xu, 2005). It provides a certain degree of confidence in assessment the clustering result because Clustering algorithms always generate clusters whether structure exists in the dataset or not. Moreover, Cluster Validation Index (CVI) is useful in answering questions like how many clusters are hidden in the data, whether it is meaningful or why clustering algorithm is chosen instead of another. This method is computational expensive by traversing throughout a range of cluster between 2 and *N*. Cluster Validation can be grouped by the approach as follows.

1. One factor approach. In the early stage, this method evaluates cluster number by within cluster dispersion. For examples, Partition coefficient (Bezdek,1974b), Partition entropy (Bezdek,1974a) as displayed in (17) and (18) respectively.

$$V_{PC} = \frac{1}{N}\sum_{j=1}^{k}\sum_{i=1}^{N}\mu_{ij}^{\ 2} \tag{17}$$

$$V_{PE} = \frac{1}{N}\sum_{j=1}^{k}\sum_{i=1}^{N}\mu_{ij}\log\mu_{ij} \tag{18}$$

Dave (1996) modified Partition coefficient by include the number of clusters to reduce the monotonic tendency.

$$V_{MPC} = 1 - \frac{k}{k-1}(1 - V_{PC}) \tag{19}$$

Kim *et al.* (2004) proposed a CVI for Gustafson-Kessel (GK) Clustering by minimization the overlap measure without considering the compactness.

2. Two factors approach. This method is widely used in most validation index which measured the two factors in term of compactness and separation. CVI can be calculated by either minimize or maximize of the function (Hubert and Schultz, 1976; Dunn, 1973; Davies and Bouldin, 1979; Rousseeuw, 1987; Halkidi et al., 2000; Halkidi et al., 2002). However these methods are designed for K-Means. In the Fuzzy version, the CVI is usually consists of two terms that measure in term of inter and intra clusters. It can be rewritten in generic function form of relevant parameters as (20) and (21) respectively.

$$InterCluster = f(v_i, v_j, k) \tag{20}$$

$$IntraCluster = f(\mu_{ij}, m, k, N, d_{ij}^{\,2}) \tag{21}$$

For examples, CVI can be defined by adding inter cluster penalty term to the intra cluster measure function (Fukuyama and Sugeno, 1989). Instead of adding the penalty term, CVI is formulated by the ratio of inter and intra cluster to improve quality in obtaining number of cluster (Xie and Beni, 1991).This ratio is extended to eliminate the monotonic tendency by adding the average distance between centroids to the compactness function (Kwon, 1998). Some CVIs are developed with specific ability to validate clustering result. For examples, CVI performs well in large variability of cluster shapes, density and number of data points. This CVI measures both compactness and separation in term of Fuzzy Covariance Matrix (Gath and Geva, 1989). CVI with ability to capture small clusters among large clusters is developed by adding the similarity between each data and centroids of dataset to improve the separation calculation (Pakhira and Bandyopadhyay, 2004, 2005). CVI works in noisy environment, is developed by subtract the Partition Coefficient with Exponential function of the ratio between compactness and separation (Wu and Yang, 2005).

Table 4 Well-known Cluster Validity Index

$V_{FS} = \sum_{j=1}^{k}\sum_{i=1}^{N} \mu_{ij}{}^{m} \left\|x_i - v_j\right\|^2 - N\left(\min_{i\neq j}\left\|v_i - v_j\right\|^2\right)$	To improve traditional approach by adding inter clustering as a penalty term.(Fukuyama and Sugeno, 1989)
$V_{XB} = \dfrac{\sum_{j=1}^{k}\sum_{i=1}^{N} \mu_{ij}{}^{m} \left\|x_i - v_j\right\|^2}{N\left(\min_{i\neq j}\left\|v_i - v_j\right\|^2\right)}$	To improve quality in obtaining number of cluster (Xie and Beni ,1991)
$V_K = \dfrac{\sum_{j=1}^{k}\sum_{i=1}^{N} \mu_{ij}{}^{2} \left\|x_i - v_j\right\|^2 + \frac{1}{k}\sum_{j=1}^{k}\left\|v_i - v_j\right\|^2}{\min_{i\neq j}\left\|v_i - v_j\right\|^2}$	To improve monotonic tendency (Kwon ,1998)
$V_{FHV} = \sum_{j=1}^{k}\left[\dfrac{\sum_{i=1}^{N} \mu_{ij}{}^{m}(x_i - v_j)(x_i - v_j)^T}{\sum_{i=1}^{N} \mu_{ij}{}^{m}}\right]^{1/2}$	To improve capability to capture large variability in cluster shapes (Gath and Geve, 1989)
$V_{PMBF} = \left[\dfrac{\left(\sum_{i=1}^{N}\left\|x_i - \bar{x}\right\|\right)\left(\max_{i\neq j}\left\|v_i - v_j\right\|\right)}{k\sum_{j=1}^{k}\sum_{i=1}^{N} \mu_{ij}{}^{m}\left\|x_i - v_j\right\|}\right]^2$	To improve capability to capture small clusters among large clusters (Pakhira and Bandyopadhyay, 2004, 2005)
$V_{PCAES} = \sum_{j=1}^{k}\sum_{i=1}^{N} \dfrac{\mu_{ij}{}^{2}}{\min_{1\leq i\leq k}\sum_{i=1}^{N} \mu_{ij}{}^{2}} - \sum_{j=1}^{k}\exp\left(\dfrac{-\min_{i\neq j}\left\|v_i - v_j\right\|^2}{\sum_{j=1}^{k}\left\|v_i - \bar{x}\right\|^2 / k}\right)$	To improve separation and ability to resist noise (Wu and Yang, 2005)
$Var = \left(\dfrac{k+1}{k-1}\right)^{1/2}\sum_{j=1}^{k}\sum_{i=1}^{N}\mu_{ij}\left[1 - \exp\left(-\left(\sum_{i=1}^{N}\left\|x_i - \bar{x}\right\| / N\right)^{-1}\right)\right] / N_k$ $Sep = 1 - \max_{i\neq u}\left(\max_{x_i}\min\left(\mu_{ij} - \mu_{uj}\right)\right), \quad V_w = \dfrac{Var/\max Var}{Sep/\max Sep}$	To improve ability to work in noisy environment (Zhang *et al.*, 2008)

Another CVI uses normalized compactness and separation to resist effect of noise. This CVI calculates compactness based on Exponential of the

similarity between each data and centroid of dataset and separation based on the degree of membership (Zhang *et al.*, 2008).

3.2.2 Model based Approach

Model based approaches are developed based on other machine learning techniques. These methods use related information other than separation and compactness of the clustering result such as the statistical gap technique that obtains number of clusters based on sum of square error of total similarity of clusters (Tibshirani et al., 2000). Later, this method is extended for Fuzzy Clustering by introducing the fuzzy term into an equation (Arima et al., 2008). An alternative approach finding number of cluster based on Gaussian distribution (Xu, 1996; Xu, 1997; Guo *et al.*, 2002).). Another approach is to incorporate the domain specific information such as identify the number of clusters from spatial information (Li and Shen, 2010), tri co-occurrence of pattern in image processing field (Koonsanit and Jaruskulchai, 2012).

3.2.3 Employ Hierarchical Clustering Procedure

This approach takes advantage of the Hierarchical clustering output or dendrogram to determine the number of clusters such as revising the clustering procedure by integrating either agglomerative or divisive into clustering procedure. The main advantage of this procedure is that clustering is not influenced by initialization seed and not falls into local minima. In addition, the number of clusters need not be specified at the beginning (Frigui and Krishnapuram, 1997). Later, this method is integrated to FCME (Li et al. ,2008). Another extension is proposed by utilizing the neighbor selection which is able to identify number of clusters even the dataset is highly overlapped (Zhang et al., 2010). Another approach is to compare the changes of total similarity within the clusters against varying of number of clusters. There is a point called “Knee” where the total similarity is slightly changes and the knee point will be obtained as optimum number of clusters (Zang and Bo, 2010).

3.3 Finding cluster initialization

Selecting the right initial centroid does not only prevent clustering get struck in the local minima but also improve the quality clustering. A poor initialization could lead to a merge cluster if a small cluster locates very close to a large cluster or divide a cluster into two clusters if the initialized centroids are from the same cluster. Normally, the random selection of data points as centroids is easy and simple for initialization but it could lead to the problem as described above. In order to overcome this problem, there are two main categories approaches to initialize the centroid as follow:

3.3.1 Partition approach

This method attempts to divide the dataset into *k* partitions and selects the initial seeds from the representation of the divided partitions. In general, the dataset is recursively divided until each partition contains small portion of data points. For examples, the dataset is divided into two sub groups until reach *k* groups based on sum of square error computation (Deelers and Auwatanamongkol, 2007). Another method divides the dataset into *k* groups randomly and selects the *k* seeds based on minimum of sum square errors (Steinley, 2003). Instead of partition the entire dataset, some studies partition attributed of data. For example, the attributes are modeled with normal distribution. Initial seeds are computed from mean and standard deviation of each attribute. (Khan and Ahmad, 2004), Another method examines the attribute of data points and recursively splits them by value of attribute into two groups (data points with attribute more than median and data with attribute less than median) until each group contains minimum number of data i.e. less than 10 data points (Redmond and Heneghana, 2007).

3.3.2 Assessment approach

This method attempts to assess the data by minimization or maximization of some mathematical function and wisely selects initial centroids from assessment function such as an augmented K-Means with a simple randomized seeding technique. The seeds are assessed based on probability function (Arthur and Vassilvitskii, 2007). LOF method can be used to select initialization seeds by computing LOF score to avoid selection outliers as seeds (Hassan *et al.*, 2009). Another assessment function measures the density of neighbor from each data point and selects seeds from potential data from high density area. (Zhang *et al*., 2010). In image segmentation, the assessment function is commonly based on image intensity; however, seeds can be selected from high density area through Expectation Maximization (EM) algorithm (Lynch, *et al.*, 2007).

MATERIALS AND METHODS

Materials

1. Hardware

    1.1 Computer Intel Core2Duo with CPU 2.4GHz
    1.2 Main memory 4GB
    1.3 Hard disk 80 GB

2. Software

    2.1 Windows XP Operating System
    2.2 Microsoft SQL Server 2005 Express Edition
    2.3 Microsoft Visual Studio 2008 Express Edition
    2.4 ASP.NET 2.0
    2.5 Eclipse

Methods

A FCM and its variants assign data points to multiple clusters but degree of membership does not well represent the degree of belonging (From Figure 5, FCM assigns data 4 to all clusters with degree of membership over zero). It causes a data point belongs to clusters more than it used to be. For example, a dataset is clustered into four clusters (C1, C2, C3 and C4) and the distance from a data point to each cluster's centroid is 15, 20, 30 and 500 respectively. In fact, this data point should not belong to C4 because its distance is too far away from the centroid. As illustrated in Figure 9 and Figure 10, this data point begins to contribute its membership to C4 when m>1.7.

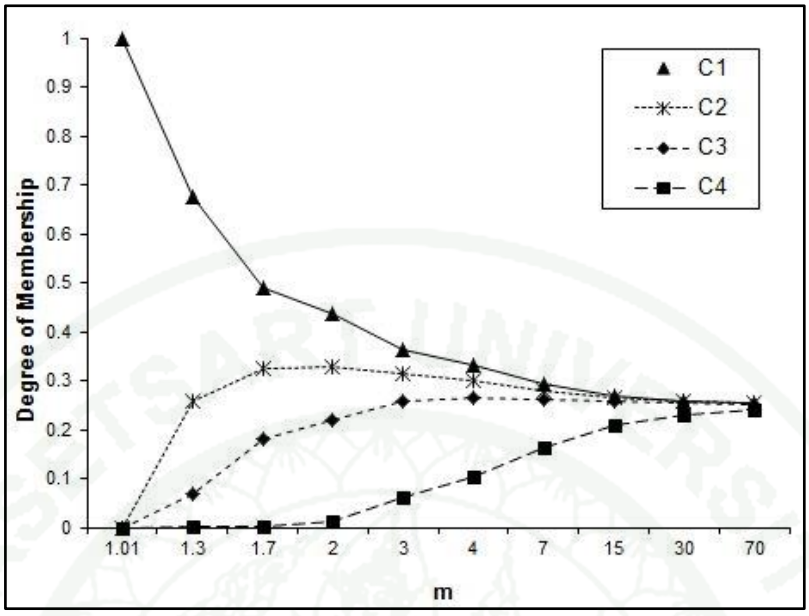

Figure 9 Impact of fuzzifier of FCM for a data point being assigned to 4 clusters (C1, C2, C3 and C4) having distance 15, 20, 30 and 500 respectively.

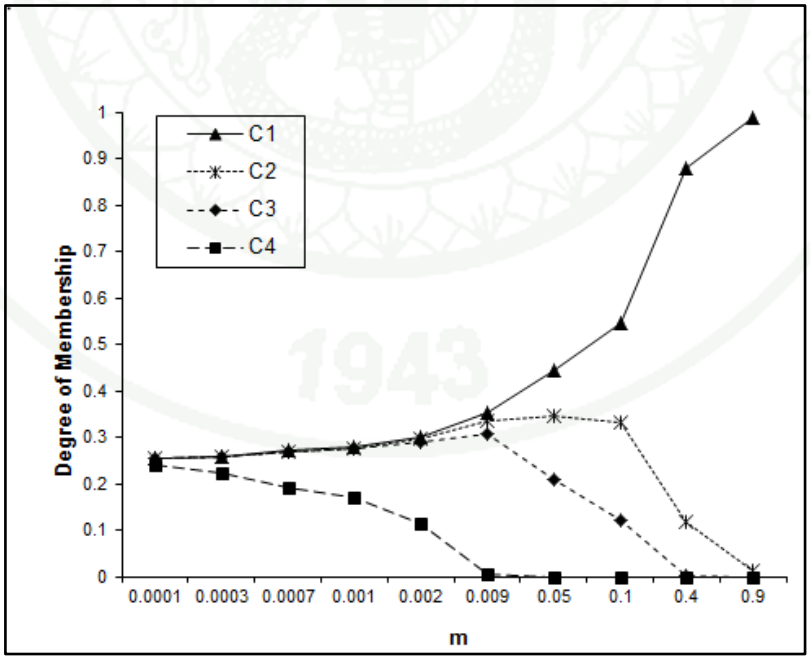

Figure 10 Impact of fuzzifier of FCME for a data point being assigned to 4 clusters (C1, C2, C3 and C4) having distance 15, 20, 30 and 500 respectively.

The larger the value of *m* is the greater contribution of irrelevant data points to the cluster. The appropriate of *m* that causes minimal distortion of this data point should be between [1.01,1.7] when this data point belongs only to C1, C2 and C3. For FCME, the graph in Figure 10 displays the mirror of FCM. The data point will be assigned to the nearest cluster or similar to K-Means Clustering when the value of *m* beyond 0.9. At the other end of the graph, FCME initially contributes this data points to the cluster C4 and stops when *m* =0.009. Thus, the range of *m* should be [0.009, 0.9] for FCME clustering.

From this example, FCME has a higher flexibility than FCM with respect to its ability to adjust fuzzifier in a wider range values. FCME has more options for the allocation of the degrees of membership of data for C1-C3. Besides, membership degree for C2 should be closer to C1 than C3. For FCM, the difference between the membership values of C1 and C2 is smallest when *m*=1.7 and the difference is 0.2. When *m* is around 0.009, FCME algorithm assigns much closer membership values for C1 and C2 than FCM. In fact, the algorithm should produce reasonable and meaningful degree of membership by discriminating unrelated data points out of the clusters. Thus, the degree of membership should represent by three types of member. In first membership type, data strongly belong to the clusters. The degree of membership should always be one. In second type, data partially belong to the cluster. The degree of membership should in between zero to one. Lastly, data should not belong to the clusters. The degree of membership should always be zero. The membership function that classifies member as described before, should be formed in Logarithmic function. In this thesis, the Exponential Fuzzy Clustering is proposed as the based algorithm. The details of Exponential Fuzzy Clustering are described as below.

1. Exponential Fuzzy Clustering

The objective function for Exponential Fuzzy Clustering (XFCM) that fulfills the above requirements, can be formulated based on two conditions. First, the condition sets μ_{ij}=0 when data do not belong to any clusters. In this case, the objective function must be zero. Second, the condition sets μ_{ij} =1 when data belongs to the cluster. The objective function must be equal to the distance function the same as FCM and FCME. The objective function that meets these conditions is shown in (22)

$$J_{XFCM} = \sum_{j=1}^{k}\sum_{i=1}^{N}(\frac{m^{\mu_{ij}}-1}{m-1})d_{ij}{}^{2}, m \in (1,\infty), \sum_{j=1}^{k}\mu_{ij} = 1. \quad (22)$$

In order to find a solution, the objective function (22) is minimized using Lagrange Multiplier. The optimum solution for degree of membership is in (23) or (24) and optimality for update centroid is in (25). (See Appendix B for the derivation)

$$\mu_{ij} = \frac{1 + k\log_m\left[\frac{1}{d_{ij}{}^{2}}\right] - \sum_{u=1}^{k}\log_m\left[\frac{1}{d_{iu}{}^{2}}\right]}{k} \quad (23)$$

Or simplify for a shorter format as (24)

$$\mu_{ij} = \frac{1 + \log_m\left[\frac{\prod_{u=1}^{k} d_{iu}{}^{2}}{(d_{ij}{}^{2})^{k}}\right]}{k} \quad (24)$$

$$v_{ij} = \frac{\sum_{i=1}^{N}[(m^{\mu_{ij}} - 1)x_{ij}]}{\sum_{i=1}^{N} m^{\mu_{ij}} - 1} \quad (25)$$

1.1 Membership Degree Enhancement

To represent the degree of membership into three types as state above, the degree of membership calculates by (23) or (24) could be presented the degree of membership by three different behaviors. It could go beyond 1 to indicate the strong relationship between data point and the cluster. It could shift below 0 to the negative number to indicate the unrelated relationship between data point and the cluster. The membership degree could be varied in range between 0 and 1 for fuzzy relation. If there is a negative degree of membership, it should have degree of membership greater than 1 so the summation of the degree of membership to one can be true. We could capture the negative degree of membership using (26).

$$(d_{ij}{}^2)^k > \prod_{u=1}^{k} d_{iu}{}^2 \quad (26)$$

Data points locate very far from the cluster are potentially assigned negative degree of membership. The negative membership degree indicates very low correlation between data and clusters. On the other hand, a data point truly belongs to the cluster if the membership degree goes beyond one. Thus, these three types of degree of membership behave as aforementioned. In addition, these properties can be used to filter out irrelevant data points when the degree of membership is negative. However, the negative degree of membership is not a Fuzzy compliance in which μ_{ij} in the range of [0,1].

In order to resolve the compliance issues, a new condition is introduced to verify cluster membership. The data points will be included in the cluster if and only if the fuzzifier parameter *m* satisfies the condition (27) otherwise data are not included by assigning $\mu_{ij}=0$ for particular clusters.

$$m \geq \frac{(d_{ij}{}^2)^k}{\prod_{j=1}^{k} d_{iu}{}^2} \tag{27}$$

Theorem 1.1 The degree of membership of XFCM is complemented to Fuzzy approach if and only if the distance of data to the cluster satisfies the condition (27).

Proof: From (24), the membership degree is greater than 0 only if

$$\log_m \left[\frac{\prod_{u=1}^{k} d_{iu}{}^2}{(d_{ij}{}^2)^k} \right] \geq -1$$

$$\frac{1}{m} \leq \frac{\prod_{u=1}^{k} d_{iu}{}^2}{(d_{ij}{}^2)^k} \tag{28}$$

$$m \geq \frac{(d_{ij}{}^2)^k}{\prod_{u=1}^{k} d_{iu}{}^2}$$

It is easily to resolve this equation and get result as (27). So if *m* is larger than the right hand side of (27), the degree of membership will always non negative number and because of $\sum_{j=1}^{k} \mu_{ij} = 1$, hence $\mu_{ij} \in [0,1]$ □

The remaining of μ_{ij} of that data to other clusters will be positive at this point and the total of μ_{ij} will be conflictive with the fuzzy clustering constraint whereas $\sum_{j=1}^{k} \mu_{ij} = 1$. Thus μ_{ij} will be normalized to become μ_{ij}^{*} as (29)

$$\mu_{ij}^{*} = \frac{\mu_{ij}}{\sum_{\mu_{ij}>0} \mu_{ij}} \tag{29}$$

In order to embed the noise filtering into algorithm, the clustering process for XFCM is modified as illustration in Figure 11.

STEP1 Predefined parameters: This step is to define the required parameters to process in the algorithm. They are; number of clusters (k), fuzzifier parameter, termination coefficient (ε) to use in termination process and initialize k centroids (v_j^0) for each cluster j.
STEP2 Compute the degree of membership: This step is to compute μ_{ij} by (23) or (24).
STEP3 Verify the degree of membership: This step verifies μ_{ij} with (27). If $\mu_{ij}<0$, set $\mu_{ij}=0$ then compute the μ_{ij}^{*} according to (29)
STEP4 Update Centroids: Once all data are allocated, the centroids get update regarding to their members.
STEP5 Validate the Termination Criteria: Repeat STEP2 until the desire condition is met then algorithm is terminated.

Figure 11 Procedure of Exponential Fuzzy Clustering.

1.2 Property of the Fuzzy exponential clustering

The algorithm tries to minimize the objective function (22) by minimization the distance within the cluster. The fuzzifier m is used to balance the weight of the degree of membership. In case of m is very large, the degree of membership is likely to be $1/k$. On the other hand if m is very close to 1, data get assigned to the single closet cluster with the degree of membership equal to 1.

Theorem 1.2 If *m* is very large and much larger than d_{ij}^2, XFCM allocates data equally to all clusters.

Proof: It is known that

$$\lim_{m\to\infty} \log_m \left[\frac{1}{d_{ij}^{\,2}} \right] \approx 0 \tag{30}$$

Thus, substitution of this equation into (24) would result μ_{ij} is 1/k for every cluster. □

Theorem 1.3 If *m* is very close to 1, XFCM assigned data to the single closet cluster with the degree of membership equal to 1.

Proof: Let define the Different Distance Product (*DDP*) as below

$$DDP = \prod_{j=1}^{k} d_{ij}^{\,2} - (d_{ij}^{\,2})^k \tag{31}$$

The maximum *DDP* is obtained from the nearest cluster and it always greater than zero. It is known that $m \in (1,\infty)$ thus at least the closet cluster satisfies condition (27). If *m* is decreased, the data begin to be filtered out from the clusters because of unsatisfactory condition (27) but at least the nearest cluster must be remained. Hence if *m* is close to 1, all other clusters would have $\mu_{ij} = 0$ but the nearest cluster would have $\mu_{ij} = 1$. □

2. Possibilistic Exponential Fuzzy Clustering

The sum of the degrees of membership across all clusters for each data point is equal to one for FCM, FCME and XFCM. These algorithms have a limitation that turns the abnormal points to be member of a cluster. In order to overcome this circumstance, possibilistic approach is proposed to relax this condition by interpreting the membership function as a degree of compatibility or possibility instead of probability constraint (Krishnapuram, 1993). As a result, PFCM represents membership degree in more reasonable and meaningful but PFCM suffers from coincident problem. In the outlier detection context, FCM, FCME, XFCM, PFCM and UPC do not well separate any potential abnormal points but minimizing their impact by assigning lower membership degrees; therefore outliers are remained in the dataset during the execution of clustering process. Thus, outliers and noises influence the centroid more or less. XFCM assigns degree of membership aggressively in term of logarithmic function by filtering the negative degree of membership out of the clusters. However XFCM is bounded with probability constraint thus data are assigned at least to one cluster. In order to develop a new algorithm that represents more meaningful degree of membership with noise filtering capability. This algorithm is proposed by combining the Possibilistic approach and Exponential Fuzzy Clustering, which is called Possibilistic Exponential Fuzzy Clustering (PXFCM). The objective function of PXFCM algorithm can be formulated as (32)

$$J_{PXFCM} = \sum_{j=1}^{k}\sum_{i=1}^{N}\frac{m^{\mu_{ij}}-1}{m-1}d_{ij}{}^{2} + \sum_{j=1}^{k}\lambda_j\left(\sum_{i=1}^{N}1-\mu_{ij}\right), m \in (1,\infty) \qquad (32)$$

where λ_j are positive number which can be computed by (37). The first term in (32) is XFCM's objective function that requires distance to the centroid as low as possible while second term forces μ_{ij} to be as large as possible. In order to derive the necessary conditions to update membership equations, it used the same approach as Krishnapuram (1993). The solution for degree of membership and updating centroid can be found in (33) (See Appendix C for the derivation).

. $$\mu_{ij} = \log_m \left[\frac{\lambda_j (m-1)}{d_{ij}^{\ 2} \ln m} \right] = \log_m \left[\frac{\lambda_j (m-1)}{\ln m} \right] + \log_m \frac{1}{d_{ij}^{\ 2}} \quad (33)$$

However, the membership degree in (33) behaves similarly to XFCM and it could out of range [0,1]. The negative μ_{ij} indicates that data should not belong to the cluster while positive μ_{ij} indicates the opposite. In order to complement the membership degree in range, a new membership degree $\mu_{ij}^{\ *}$ is defined as (34)

$$\mu_{ij}^{\ *} = \begin{cases} 0 & \text{if } \mu_{ij} \le 0 \\ \mu_{ij} & \text{if } \mu_{ij} > 0 \end{cases} \quad (34)$$

The negative degree of membership will set to 0 in (34). The remaining degree of membership will be normalized by the recalculation in (35)

$$\mu_{ij}^{\ *} = \frac{\mu_{ij}}{\sum_{j=1}^{k} \mu_{ij}} \quad (35)$$

To find the optimal of updating centroid equation (See Appendix C for the derivation), the result is the same equation as XFCM in (24). But it uses $\mu_{ij}^{\ *}$ in (34) and (35) as μ_{ij} replacement during the centroid computation. The new update centroid can be found as (36)

$$v_{ij} = \frac{\sum_{i=1}^{N} [(m^{\mu_{ij}^{\ *}} - 1) x_{ij}]}{\sum_{i=1}^{N} m^{\mu_{ij}^{\ *}} - 1} \quad (36)$$

In general, the value of λ_j represents the relative degree of the second term in objective function (32). If the two terms are weight roughly equal, then λ_j should be the order of $d_{ij}^{\ 2}$ which illustrates in (37)

$$\lambda_j = K \frac{\sum_{i=1}^{N} \left(\frac{m^{\mu_{ij}^*} - 1}{m-1} d_{ij}^{\,2} \right)}{\sum_{i=1}^{N} \left(\frac{m^{\mu_{ij}^*} - 1}{m-1} \right)} \tag{37}$$

where K is the adjustable weight which typically be one. The estimated value of λ_j in (37) is based on intra cluster distance. The larger value of λ_j produces higher membership degree (33) but there is a penalty by the second term in case of the distance is very high. The value of λ_j should be a fix value for all clusters during the clustering execution. If λ_j is varied, it may leads to instabilities such as objective function does not minimized as pointed out by Krishnapuram (1993). In this thesis's experiments, they will be fixed throughout the process. The λ_j in (37) is calculated before PXFCM is processed. Thus μ_{ij} can be computed that based on the degree of membership of XFCM in (29).

2.1 Outlier Detection in PXFCM

The membership degree (33) can go down below zero ($\mu_{ij} < 0$) if a distance value is over a certain limit which is captured by (38) (See Theorem 2.1 for the proof).

$$d_{ij} < \frac{\lambda_j (m-1)}{\ln m} \tag{38}$$

Theorem 2.1 The degree of membership of PXFCM is non-negative if and only if the distance of data to the cluster satisfies the condition (38).

Proof: From (33), the membership degree is greater than 0 only if

$$\log_m\left[\frac{\lambda_j(m-1)}{{d_{ij}}^2 \ln m}\right] > 0$$
$$\frac{\lambda_j(m-1)}{{d_{ij}}^2 \ln m} > 1 \qquad (39)$$
$${d_{ij}}^2 < \frac{\lambda_j(m-1)}{\ln m}$$

It is easily to resolve the (33) and get result as shown in(38). So if the d_{ij}^2 is larger than the right hand side of (38), the degree of membership will always non negative number. □

In case the condition (38) is failed, PXFCM recalculates degree of membership in term of μ_{ij}^* as illustrated in (34). If the summation of μ_{ij}^* of a data point across all clusters is zero, that data point does not belong to any clusters and it is an outlier or (40) is true.

$$\sum_{j=1}^{k} {\mu_{ij}}^* = 0 \qquad (40)$$

On the other hand, PXFCM assigns higher degree of membership for data points locate near the centroid (i.e. d_{ij}^2 is low). These data points have less probability to be determined as outliers. Basically, a data point is considered as outlier if it does not locates within boundary of any clusters in the dataset. PXFCM uses condition (38) to determine the boundary of each cluster. The right term of (38) relies on m which is a constant throughout the clustering execution and λ_j which calculated by (37) is a specific variable for each cluster. A data point is assigned to the cluster j if its distance locates within the boundary i.e. (38) is true.

In order to benchmarking with other outlier techniques, It is necessary to define an outlier parameter, Exponential Outlier Factor (XOF), and XOF is defined from the overall residual distance from all clusters as illustrated in (41). The outlier data will have the higher XOF value.

$$XOF_i = \sum_{j=1}^{k}\left(d_{ij}{}^2 - \frac{\lambda_j (m-1)}{\ln m} \right) \quad (41)$$

By using PXFCM, there are several advantages over other fuzzy clustering algorithms. Firstly, the algorithm is embedded with outlier detection which handles the outliers by assigning zero degree of membership to those data points. Secondly, when a data point has degree of membership below zero or over one, recalculation of degree of membership is carried out. Thus, the μ_{ij}^{*} indirectly involves the distance to other clusters and coincidence clusters can be prevented. Finally, the algorithm processes based on single parameter, hence optimization does not complicate. PXFCM algorithm can be operated using procedure as Figure 12.

STEP1 Predefined parameters: This step is to define the required parameters to process in the algorithm. They are; number of clusters (k), fuzzifier parameter, termination coefficient (ε) to use in termination process and initialize k centroids (v_j^0) for each cluster j.
STEP2 Compute the degree of membership: This step is to compute μ_{ij} by (33).
STEP3 Verify the degree of membership: This step verifies μ_{ij} with (38). If $\mu_{ij}<0$, set $\mu_{ij}=0$ then compute the μ_{ij}^{*} according to (29)
STEP4 Update Centroids: Once all data are allocated, the centroids get update regarding to their members.
STEP5 Validate the Termination Criteria: Repeat STEP2 until the desire condition is met then algorithm is terminated.

Figure 12 Procedure of Possibilistic Exponential Fuzzy Clustering.

2.2 Property of fuzzifier

The role of parameter m is equivalent to fuzzier m in FCM and XFCM that controls the fuzziness level of algorithm. The one dimension data from Section 3.1 is re-explained again. PXFCM produces degree of membership as illustrated in Figure 13. PXFCM strongly indicates membership for the data locate near centroid by producing the degree of membership over 1. On the other hand, PXFCM stops

assigning data to the cluster when data locates too far away. For m=1.1, only the data with very close to the centroid will get positive degree of membership (in this case, data with distance below 0.4) as indicated in Figure 13. When fuzziness level is increased (m is increased), PXFCM assigns more data to the cluster i.e. number of data in the grey area is decreased. In addition, PXFCM reduces to K-Means when m is very close to 1 and PXFCM produces μ_{ij} to 0.5 for all i,j when m is very large.

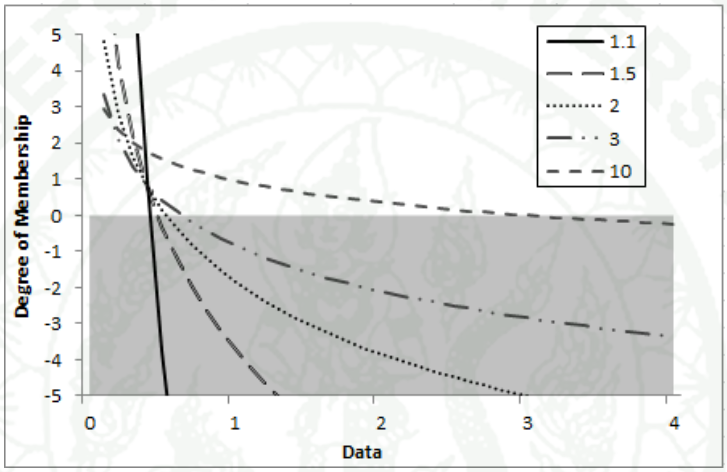

Figure 13 Degree of membership produced from PXFCM for 1 dimension data.

From (38), the meaning of this equation is not only used to control data assignment. PXFCM can be optimized for partition quality and noise filtering. In case of its known priori that dataset does not contain abnormal data points, PXFCM should be executed without noise filtering or all data should be allocated at least to a cluster. It is true only if the largest distance of data in the dataset is less than the function of λ_j as illustrated by (42).

$$\max d_{ij}^{\,2} < \frac{(m-1)}{\ln m} \min \lambda_j \qquad (42)$$

This equation is likely to be true if m value is large. On the other hand, if m is small enough (i.e. 1.5 or 2 as most common choice for FCM), PXFCM operates with noise filtering. This is useful in case the dataset contains noisy data and outliers.

2.3 Illustration of membership computation with PXFCM

Suppose a dataset consists of 2 dimensions data as follow (1,1), (2,2) and (6,6) and (20,20). These data points will be clustered into 2 clusters. Assume there is two initialized centroids (1.5,1.5) and (2.5,2.5) for both clusters. In this example, m=1.5 is used for the calculation.

Initially, similarity of each data point and centroids are precomputed. Next, the initial μ_{ij} is calculated based on XFCM by (43) and normalized by (29). This initial μ_{ij} will be used to calculate λ_j . Then, λ_j is computed using (37). λ_j for Cluster #1 is 0.5 and λ_j for Cluster #2 is 17.05. Finally, μ_{ij} is calculated based on PXFCM by (27) and positive normalized μ_{ij} by (44). All results are summarized in Table **5** and Table 6.

Table 5 Example of calculation using PXFCM for Cluster #1

Data	Similarity	μ_{ij}(XFCM)	μ_{ij}^*(XFCM)	μ_{ij}(PXFCM)	μ_{ij}^*(PXFCM)
(1,1)	0.5	3.21	1	14.1	0.59
(2,2)	0.5	0.5	0.5	14.1	0.5
(6,6)	40.5	-0.12	0	3.26	0.42
(20,20)	684.5	0.36	0.36	-3.71	0

Table 6 Example of calculation using PXFCM for Cluster #2

Data	Similarity	μ_{ij}(XFCM)	μ_{ij}^*(XFCM)	μ_{ij}(PXFCM)	μ_{ij}^*(PXFCM)
(1,1)	4.5	-2.21	0	9.74	0.41
(2,2)	0.5	0.5	0.5	14.1	0.5
(6,6)	24.5	1.12	1	4.5	0.58
(20,20)	612.5	0.64	0.64	-3.44	0

The above example shows that XFCM divides μ_{ij} into three types as indicated above. XFCM assigns (1,1) to only Cluster#1 and assigns (6,6) to only

Cluster#2. (2,2) and (20,20) partially belong to both clusters. In this example data (20,20) belongs to both clusters when calculates with XFCM while it is detected as outlier when using PXFCM.

3. Automate algorithm with Agglomerative Fuzzy Clustering

Agglomerative clustering starts with each data point as a cluster. This clustering method forms the nested clusters by successively merging clusters. The output of this method returns into tree structure of the data which is called *dendrogram*. The dendrogram is analyzed by selecting the threshold and cut the tree at a suitable level to identify the clusters. However with large dataset, the dendrogram is impractical due to the high complexity of similarity computation. Also if the number of clusters is small, the computational complexity needs to complete the tree and can be expensively.

In general, Fuzzy clustering and its variants are similar in term of implementation. These Fuzzy clustering methods begin with the setting of fuzzifier and number of clusters as input parameters. The process continues by iterative execution and produces degree of membership and centroids. There are two main advantages of Agglomerative Fuzzy Clustering (AFC). The number of clusters is obtained during the execution and clustering is not influenced by initialize and local minima. This method begins with specifying the number of cluster larger than the true number of clusters. Then, those clusters with total degrees of membership lower than a specify threshold will be merged. Nevertheless, small clusters could be merged to the larger clusters (Frigui and Krishnapuram, 1997). In order to prevent the merge of small clusters, in this thesis the centroid similarity ($\delta(v_j,v_k)$) is proposed by comparing data attributes (q where $q=1...M$ and M is the number of dimensions) against threshold (θ^2Q where θ is [0,1]) as (45). In case of multiple centroids can be merged ($\delta(v_j,v_k)< \theta^2Q$), the new centroid is simply computed by an average of each attribute. From experiments, θ sets to 0.1-0.3 is generally used (Li *et al.*, 2008).

$$\delta(v_{j,}v_k) = \sum_{q=1}^{M}\left(\frac{v_{jq} - v_{kq}}{\max(v_{jq}, v_{kq})}\right)^2 \quad (45)$$

Basically, every fuzzy clustering algorithm has a target result indirectly indicating by fuzzifier. As illustrated in previous section in Figure 5 to Figure 8, the result in each clustering could be the same as K-Means or produced average degree of membership i.e. Fuzzifier does not only represent the level of fuzziness, but also represents how the target clustering will be. The fuzzifier is depended on algorithm. High or low value of fuzzifier causes the distribution of degree of membership (*Var*) in different behaviors. Unfortunately, existing CVIs do not take fuzzifier validation into account. From Figure 14, value of V_{XB} should minimize in order to represent a good clustering result while Normalized Variance (*NVar*) should maximize to avoid degree of membership to be average (the degrees of membership become *1/k*). Hence, the new index (V_{XB}^{*}) that capture fuzzifier should be tradeoff between these terms and new CVI that capture fuzzifier is defined as (46)

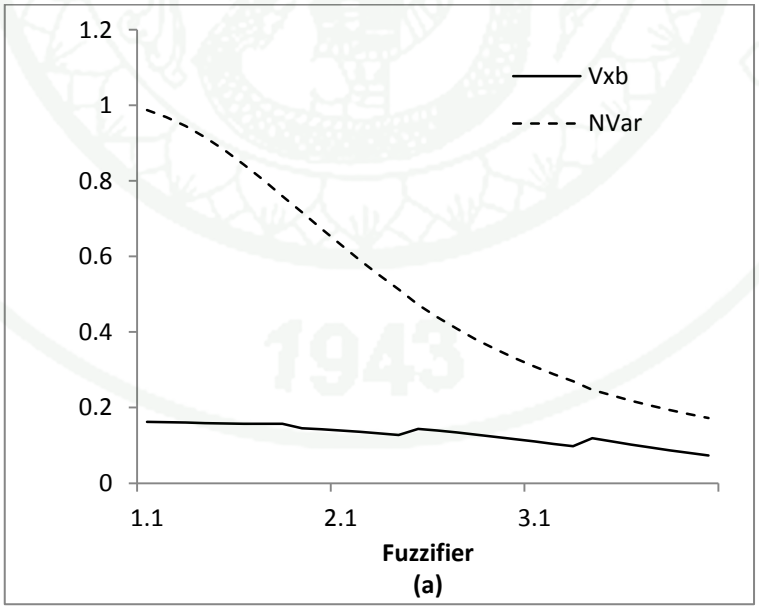

Figure 14 Changing of V_{XB} and *NVar* against fuzzifier from IRIS dataset when clustering by FCM with fuzzifier from 1.1-4.0.

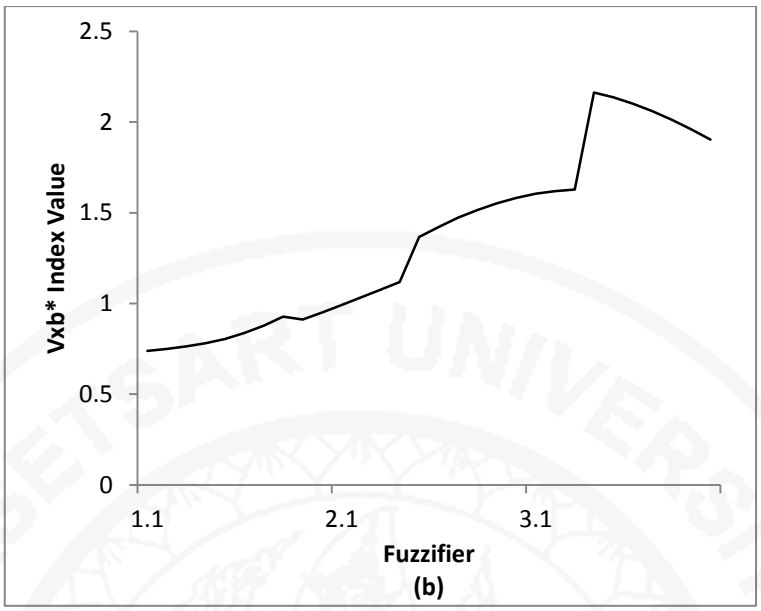

Figure 15 Tradeoff result of V_{XB} and $NVar$ against fuzzifier (V_{XB}^{*}) from IRIS dataset when clustering by FCM with fuzzifier from 1.1-4.0.

$$
\begin{aligned}
V_{XB}^{*} &= NVar * V_{XB}, \; NVar = \frac{Var}{MaxVar} \\
Var &= \frac{\sum_{j=1}^{k}\sum_{i=1}^{N}\left(\mu_{ij} - \overline{\mu_{ij}}\right)^2}{Nk-1}, \; \overline{\mu_{ij}} = \frac{\sum_{j=1}^{k}\sum_{i=1}^{N}\mu_{ij}}{Nk} \\
MaxVar &= \frac{N\left(1-\frac{1}{k}\right)^2 + \frac{Nk-N}{k^2}}{Nk-1} = \frac{N(k-1)}{k(Nk-1)}
\end{aligned}
\tag{46}
$$

Step 1: Input k, θ, m=1.1 and ε (Terminate coefficient).
Step 2: Execute Agglomerative Fuzzy Clustering.
- Compute similarity between data and centroids.
- Compute degree of membership of each data against centroids.
- Compute centroids based on degree of membership.
- Compute centroid similarity as (45) and compare with threshold.
- Merge centroid and update the number of clusters.
- Repeat Step 2 until Terminate condition is met.
Step 3: Obtain optimum fuzzifier
- Set m=m+0.1, k=optimum number of clusters from Step 2.
- Perform normal Fuzzy Clustering
- Calculate V_{xb}^{*} as (46)
- Repeat Step 3 and return output if V_{xb}^{*} begins to drop.

Figure 16 Procedure of Generalized Agglomerative Fuzzy Clustering.

The new AFC procedure consists of three steps as illustrated in Figure 16. In step 1, the algorithm required k to be larger than actual number of clusters, initial fuzzifier to any number greater than 1 (e.g. m=1.1) and termination condition to a small number (e.g. ε=1). In step 2, normal fuzzy clustering is executed and clusters are merged according to centroid similarity at the end of iteration. This step ensures that the optimum number of clusters is obtained. In step 3, m is increased and process fuzzy clustering until value of V_{XB}^{*} is decreased (see Figure 15). This step ensures that the optimum fuzzifier is obtained.

In general, this procedure can be applied to any fuzzy clustering; this procedure requires only one parameter.

RESULTS AND DISCUSSION

Results

In this study, the experiments were divided to validate performance regarding to the three main objectives of this thesis. Firstly, XFCM was validated the performance in term of representation of the degree of membership. In these experiments, XFCM were adapted to predict the ratings for Collaborative Filtering domain whereas clustering quality is crucial to achieve high accuracy of prediction. Secondly, PXFCM was validated the clustering quality by the centroid errors and Rand index on various dataset as well as benchmarking the outlier detection performance with well-known algorithms. Lastly, AFC was applied to FCM and XFCM to validate the performance against original FCM and XFCM. In addition, the optimum fuzzifier produced from AFC, were compared against an estimated equation on various dataset.

1. Performance of XFCM

The experiments were setup on the two real datasets from the MovieLens website. The first dataset was collected from user rating during the seven-month period from September 19th, 1997 through April 22nd, 1998 by GroupLens Research Project at the University of Minnesota. It consists of 100,000 ratings made by 943 users on 1,682 movies. The dataset is very sparse with 0.9396 sparsity level. Such level measures by 1-(nonzero entries/total entries). Additionally, actual number of clusters is unknown. The second dataset contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in the year 2000. This dataset has a 0.9640 sparsity. The method to compute sparsity is calculated using the same method as 100K MovieLens Dataset.

These two datasets were mainly used in collaborative filtering domain. Collaborative Filtering (CF) is a technique using in recommendation system and is one

of the most popular techniques because of its simplicity and ease of use. CF predicts the interest of a user by collecting and using information from past users who have the same opinions. By nature of CF, the dataset consists of a lot of missing values. These missing values prevent CF methods to find the users with the same opinions. With clustering method, missing values are handled by calculate similarity against centroids which always no missing value. To benchmark the CF methods, the dataset usually divides into two sets. First set is divided to build the model by learning the actual ratings. Second set is for testing which the built model calculates ratings for comparison. The difference between computed and actual ratings is compared in term of error measurement such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), etc. In these experiments, clustering results were validated using MAE for both fuzzifier and number of clusters.

The most common approach for CF is based on neighborhood models that attempt to provide recommendation by either user-user approach (Herlocker, *et al.*, 1999) or item-items approach (Sarwar , *et al.*, 2001; Karypis, 2001; Linden, *et al.*, 2003). User-user based methods predict user rating from users with similar preferences and item-item based methods predict user rating from ratings made by same user on similar items. Item-item based method is more favorable due to users are more familiar with items previously preferred by them rather than other users with the same preferences.

In general, matrix factorization methods (Koren and Bell, 2011; Vozalis, *et al.*, 2009; Sarwar , *et al.*, 2000; Randle, 2010) are the most popular in this research area. These methods try to map large users and items matrix in the lower dimensions using latent factor. They produce more accurate than neighborhood models because these methods optimize prediction based on global ratings while neighborhood models compute ratings based on local neighbors (Koren and Bell, 2011). However, many neighborhood models are preferred in many well-known recommendation systems like Amazon (Linden, *et al.*, 2003), TiVo (Ali and Stam, 2004) due to their simplicity and ease of use.

In this experiment XFCM was benchmarked with Item-based method (Sarwar , *et al.*, 2001) and matrix factorization (SVD).

1.1 Item based method

The basic idea of Item-based is trying to find similar items. Similarity computation is performed between two items by first isolate the users who have rated both of these items then perform similarity computation to determine the similarity. Item-based prediction ($P_{u,i}$) is calculated based on most *N* similar items by (47)

$$P_{u,i} = \frac{\sum_{all\ items\ N\ similar\ to\ item\ i} (S_{i,N} \cdot R_{u,N})}{\sum_{all\ items\ N\ similar\ to\ item\ i} S_{i,N}} \tag{47}$$

where ($S_{i,N}$) is similarity measure of all similar items *N* and item *i* and $R_{u,N}$ is given rating score by user *u* to items *N*. There are numerous ways of using the similarity calculation technique such as Adjusted Cosine in (48) or Pearson Coefficient in (49)

$$sim(i,j) = \frac{\sum_{u \in U} (R_{u,i} - \overline{R_u})(R_{u,j} - \overline{R_u})}{\sqrt{\sum_{u \in U} (R_{u,i} - \overline{R_u})^2} \sqrt{\sum_{u \in U} (R_{u,j} - \overline{R_u})^2}} \tag{48}$$

$$sim(i,j) = \frac{\sum_{u \in U} (R_{u,i} - \overline{R_i})(R_{u,j} - \overline{R_j})}{\sqrt{\sum_{u \in U} (R_{u,i} - \overline{R_i})^2} \sqrt{\sum_{u \in U} (R_{u,j} - \overline{R_j})^2}} \tag{49}$$

where *sim(i,j)* is similarity of each item *i* to cluster *j*. $R_{u,i}$ and $R_{u,j}$ is a given rating score by user *u* to item *i* and *j* respectively, $\overline{R_u}$ is average rating of user *u*.

1.1 Matrix Factorization (SVD)

SVD is a technique that decomposition a matrix M_{mxn} into three sub-matrices as illustrated in (50)

$$M_{mxn} = U_{mxr} \sum_{rxr} V_{rxn}^{T} \tag{50}$$

Where U_{mxr} and V_{rxn} are two orthogonal matrices and Σ_{rxr} is a diagonal matrix. SVD is a useful technique especially to reduce the dimension of data by providing the best lower rank approximation in term of Frobenius norm (Sarwar , *et al.*, 2000). SVD reduces the matrix size by determining the largest *k* diagonal value of matrix Σ_{rxr} and reduces matrix U_{mxr} and V_{rxn} accordingly. Then reconstruct the matrix M_k from (51). The prediction is then computed using (52)

$$M_k = U_k \sum_k V_k^{T} \tag{51}$$

$$R_{u,i} = \overline{R_u} + U_k \sqrt{\sum_k}^{T} (r) + \sqrt{\sum_k} V_k^{T} (c) \tag{52}$$

where *r* is r^{th} row of $U_k \sqrt{\sum_k}^{T}$, *c* is c^{th} column of $\sqrt{\sum_k} V_k^{T}$ and $\overline{R_u}$ is the average ratings of user.

1.2 Item based Fuzzy Clustering

The idea of item based fuzzy clustering is to adapt clustering algorithm over Item based method. This is done by replacing the distance computation in

Clustering algorithm with similarity (48). Then the prediction is calculated based on product of similarity and centroid as (53)

$$R_{u,i} = \sum_{j=1}^{k} \mu_{ij} \cdot v_j \tag{53}$$

where $R_{u,i}$ is a predict rating of user u on item i. v_j is a centroid of cluster j and μ_{ij} is the degree of membership of item i to cluster j. By doing this, time complexity to compute is reduced from $O(N)$ to $O(k)$ when calculate prediction comparing to item based method.

1.3 Benchmarking Results

The experiments were performed on two MovieLens datasets by separation both dataset 80% for learning the model and 20% for testing. Errors were measured by Mean Absolute Error (MAE) as illustrated in (54)

$$MAE = \frac{\sum_{i=1}^{n} |p_i - q_i|}{n} \tag{54}$$

where p_i is prediction value, q_i is actual rating made by user and n is total number of prediction value. In term of benchmarking with fuzzy clustering, clustering method is better if the quality of prediction is high with low MAE. To achieve this, the data to be assigned to clusters truly should be a member of the cluster. However the prediction equation in CF dataset in (47), (52) and (53), they rely on the neighbor. If these neighbors do not reflect the actual relationship with the prediction, it would difficult to get high prediction accuracy. In clustering perspective, if the data do not

truly belong to the cluster but clustering algorithm assigns them to the clusters, the prediction that calculates by (53) will be overfitted from unrelated data.

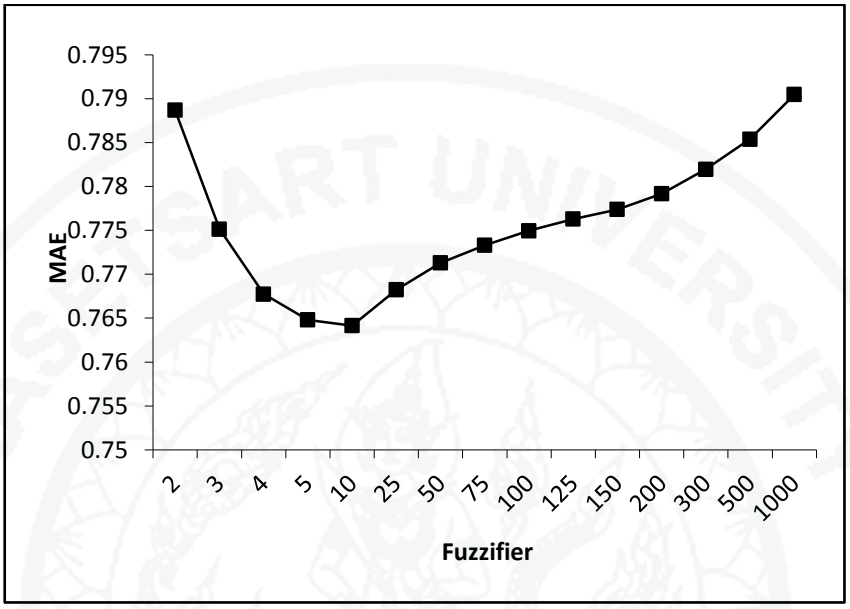

Figure 17 MAE measurement for 100K MovieLens when clustering with XFCM at variation of *m*.

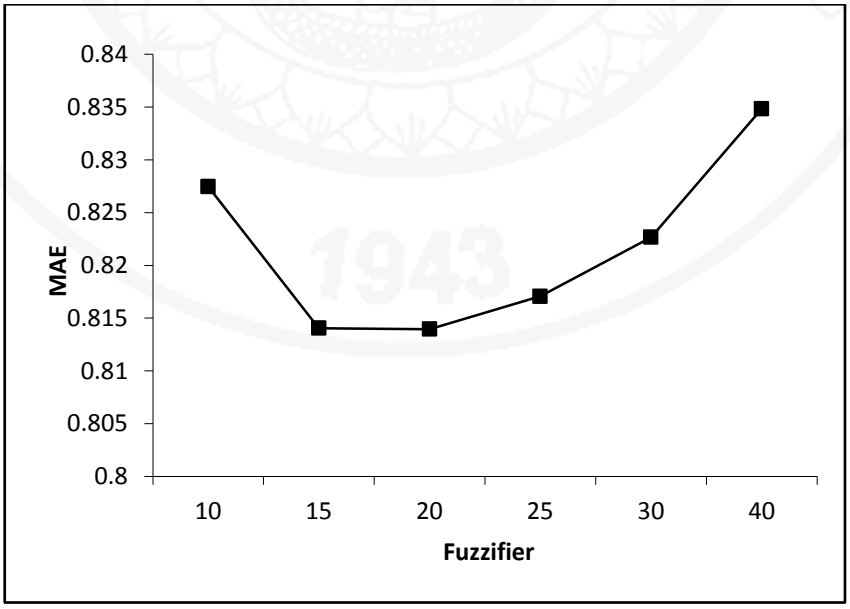

Figure 18 MAE measurement for 100K MovieLens when clustering with FCME at variation of *m*.

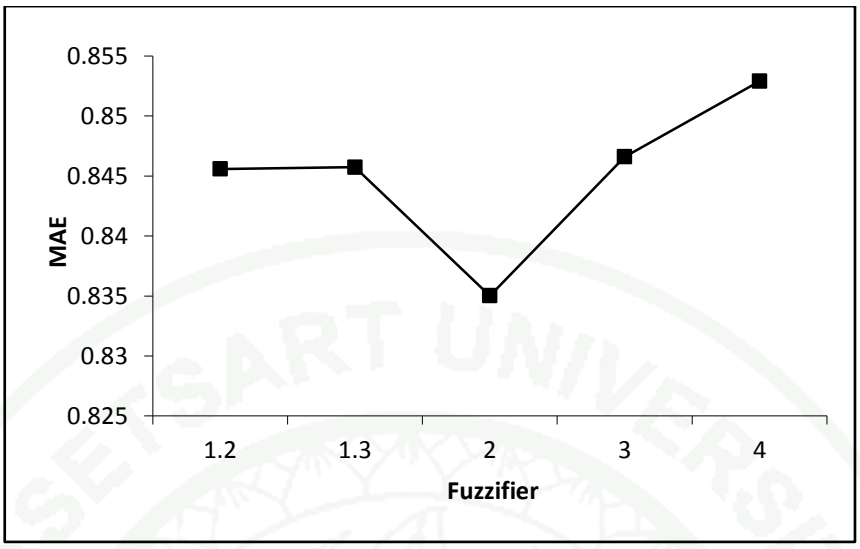

Figure 19 MAE measurement for 100K MovieLens when clustering with FCM at variation of *m*.

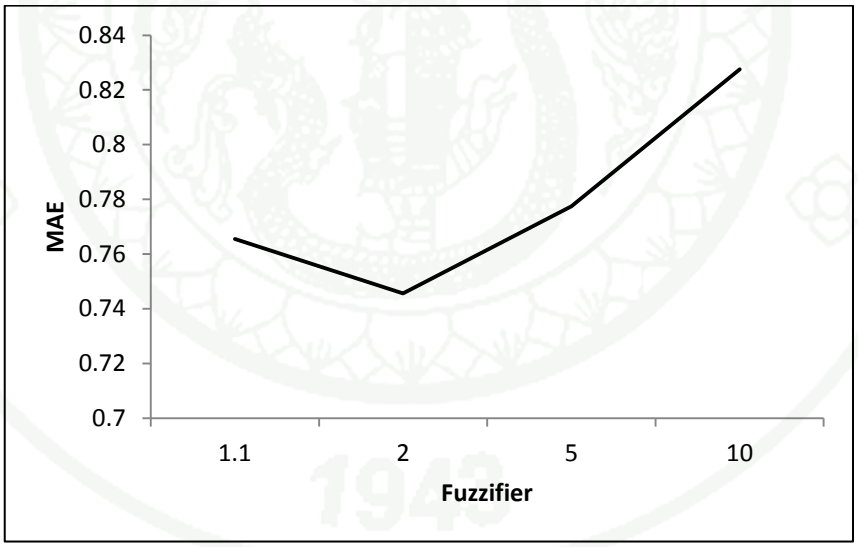

Figure 20 MAE measurement for 1M MovieLens when clustering with XFCM at variation of *m*.

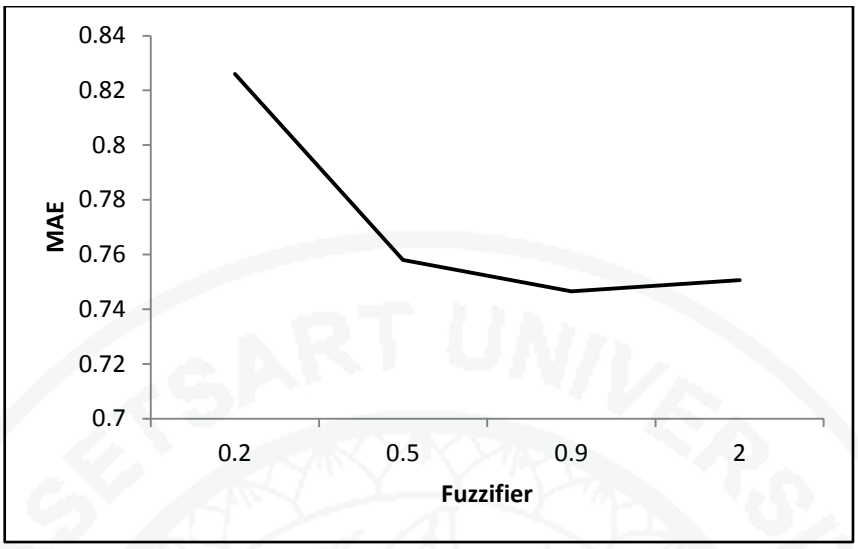

Figure 21 MAE measurement for 1M MovieLens when clustering with FCME at variation of *m*.

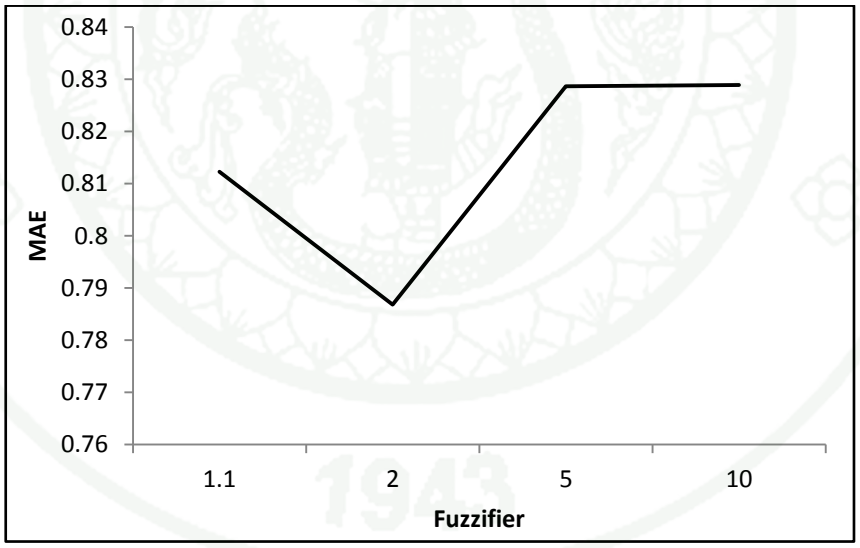

Figure 22 MAE measurement for 1M MovieLens when clustering with FCM at variation of *m*.

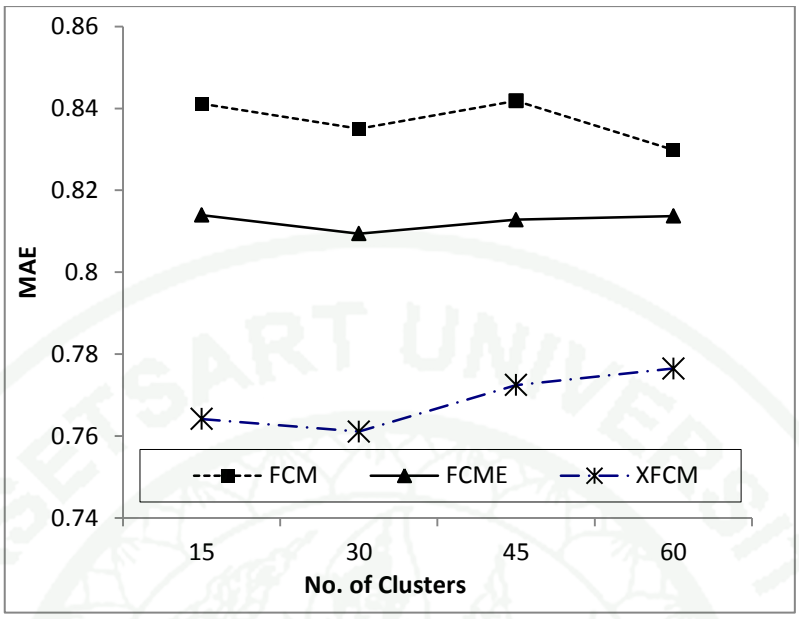

Figure 23 MAE measurement for 100K MovieLens when clustering with FCM, FCME and XFCM at variation of number of clusters.

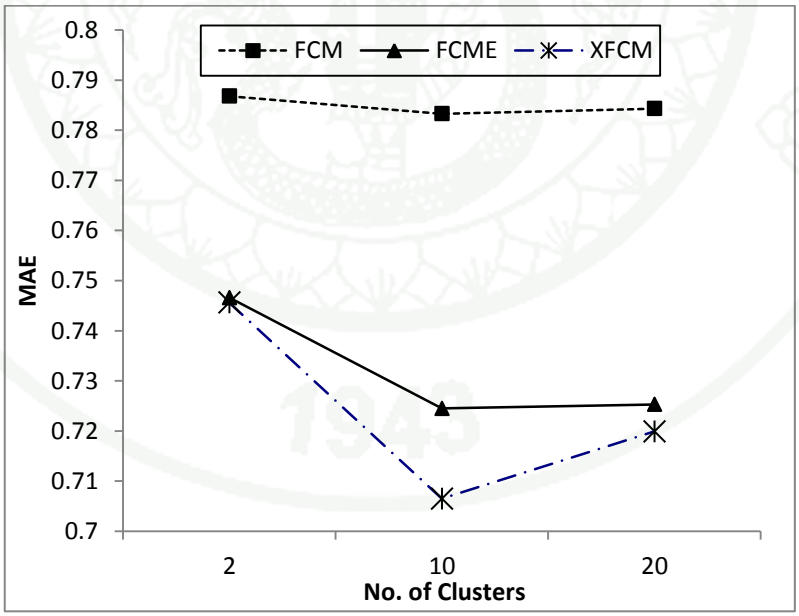

Figure 24 MAE measurement for 1M MovieLens when clustering with FCM, FCME and XFCM at variation of number of clusters.

In order to perform clustering, fuzzifier (*m*) was estimated by clustering data at different value of *m*. The curve shown in Figure 17 to Figure 22 is U-shaped curve which binary search can be used to find the lowest points of the curve. The best result was at *m*=10 for 100K MovieLens and *m*=2 for 1M MovieLens when using XFCM, *m*=20 for 100K MovieLens and *m*=0.9 for 1M MovieLens when using FCME and *m*=2 for 100K MovieLens and *m*=2 for 1M MovieLens when using FCM. These fuzzifier values yielded the lowest MAE and they were used for further experiments.

For number of clusters, FCM, FCME and XFCM were executed again with various numbers of clusters. The optimum numbers of clusters were 30 and 10 clusters for 100K MovieLens and 1M MovieLens respectively as illustrated in Figure 23 and Figure 24.

The MAE measured from optimum result of FCM, FCME and XFCM for both MovieLens datasets were benchmarked with Item based method and SVD. The results showed that XFCM yielded the best result with the lowest MAE on 100K MovieLens and 1M MovieLens as illustrated in Figure 25 and Figure 26 respectively.

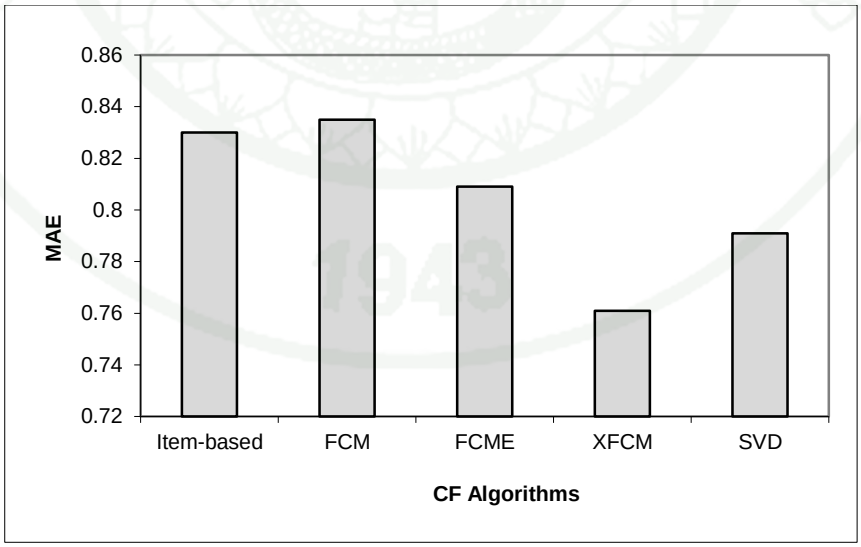

Figure 25 Benchmarking Result between FCM, FCME and XFCM based CF with Item based method and SVD for 100K MovieLens Dataset.

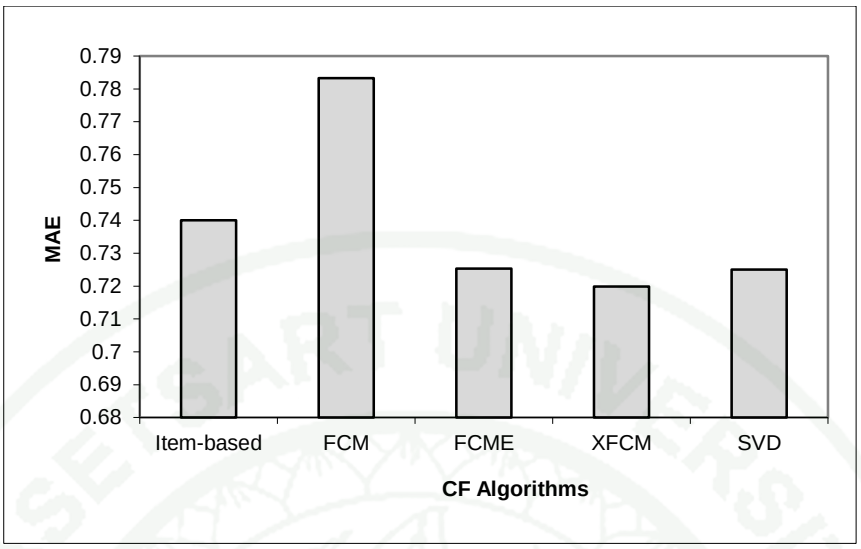

Figure 26 Benchmarking Result between FCM, FCME and XFCM based CF with Item based method and SVD for 1M MovieLens Dataset.

The distribution of μ_{ij} from the best results generated by each clustering algorithms were analyzed as illustration in Table 7. S.D. (Standard Deviation) of μ_{ij} was in similar range while a large number of μ_{ij} =0 was generated by XFCM. This meant XFCM used much lower number of data to update the centroid (around 50%). Thus the centroid was not overfitted with the noise.

Table 7 Average S.D. of Membership Degree for 100K and 1M MovieLens Dataset

	FCM	FCME	XFCM
100K MovieLens Dataset			
Total Number	24,750	24,750	24,750
S.D.	0.1029	0.1150	0.0911
Count No. of $\mu_{ij}=0$	0.02%	1.50%	50.98%
1M MovieLens Dataset			
Total Number	36,820	36,820	36,820
S.D.	0.0493	0.1306	0.2143
Count No. of $\mu_{ij}=0$	0.00%	5.89%	71.95%

The Standard Deviation (S.D.) of each algorithm was not very different from each other while a large number of $\mu_{ij} = 0$ were generated by XFCM as a result of aggressive membership distribution. Both FCME and FCM generate ratings from influenced centroids. Thus, it was difficult to achieve high accuracy using these methods. Moreover, FCM spread the membership of all data to every cluster as illustrated in Table 8 and Table 9.

Table 8 The Degree of Membership of Movie ID #1000 from 100K MovieLens dataset

Cluster No.	FCM	FCME	XFCM
1	0.1236	7.2E-07	0
2	0.0712	0.0002	0.1872
3	0.2538	0.0001	0.0543
4	0.1097	8.16E-06	0
5	0.0145	5.22E-05	0.2301
6	0.0050	2.26E-05	0
7	0.1063	0.9316	0.1529
8	0.0068	1.46E-05	0
9	0.0500	0.0543	0
10	0.0272	3.4E-06	0.3754
11	0.0924	0.0028	0
12	0.0015	2E-08	0
13	0.0206	0.0097	0
14	0.0116	0.0004	0
15	0.1050	0.0006	0

Table 9 The Degree of Membership of Movie ID #1000 from 1M MovieLens dataset

Cluster No.	FCM	FCME	XFCM
1	0.1167	0.1557	0.3779
2	0.0961	0.0933	0
3	0.1222	0.2202	0.4804
4	0.0580	0.0091	0
5	0.1014	0.0195	0
6	0.0922	0.0619	0
7	0.1170	0.1712	0.1378
8	0.1039	0.1395	0.0039
9	0.0797	0.0443	0
10	0.1128	0.0852	0

The ratings calculated from (53) were eventually overwhelmed by every rating in the dataset as mentioned earlier. For FCME, the membership distribution looked similar to K-Means clustering since only Cluster 7 is highly correlated to the Movie ID #1000 for 100K MovieLens dataset. For 1M MovieLens dataset, the data strongly correlated to clusters 1,3,7 and 8. Although the contribution to irrelevant clusters of Movie ID #1000 was low, the prediction could be deviated by other data in the dataset. For XFCM, only five and four relevant clusters are used to compute ratings for Movie ID #1000 on 100K MovieLens dataset and 1M MovieLens dataset respectively. Thus, the centroids were computed only using relevant data. In CF perspective, not all centroids were used to predict ratings of a user for Movie ID #1000 as indicated in Table 8 and Table 9. Thus, the ratings computed using XFCM based CF did not overfit.

2. Performance of PXFCM

In this section, the experiments divided into two issues to evaluate performance of PXFCM. The first issue verified the clustering quality and the second issue

validated ability of the outlier detection. One small dataset, two synthetic datasets and three real public datasets from UCI (BUPA liver disorders, IRIS and Wine), were applied in these experiments. All algorithms used the same parameter settings such as number of clusters, initial centroids and stop conditions. There were five Fuzzy Clustering algorithms with one parameter to benchmark with PXFCM i.e. FCM, PFCM, XFCM and UPC. PXFCM used different fuzzier value in some cases.

2.1 Clustering Quality

The first experiment was evaluated on 2 synthetic datasets and X16 (Wachs, 2004). First dataset (E4) consists of 4 clusters with 46 instances in each clusters. This dataset was generated by uniform distribution within the boundary of non-overlapping of 4 ellipse shapes which defined as (55)

$$\begin{aligned}
&\frac{(x-11)^2}{12.25}+\frac{(y-11)^2}{9}=1\\
&\frac{(x-4)^2}{9}+\frac{(y-4)^2}{9}=1\\
&\frac{(x-11)^2}{4}+\frac{(y-3.3)^2}{9}=1\\
&\frac{(x-3.5)^2}{9}+\frac{(y-11)^2}{9}=1
\end{aligned} \tag{55}$$

Second dataset (E2) consists of 2 clusters with 101 instances in each cluster. The data were uniformly selected within the boundary of two overlapped clusters which defined as (56).

$$\begin{aligned}
&\frac{(x-10)^2}{49}+\frac{(y-10)^2}{9}=1\\
&\frac{(x-4)^2}{9}+\frac{(y-9)^2}{49}=1
\end{aligned} \tag{56}$$

The true centroid of each cluster was the center of ellipse which are (11,11), (4,4), (11,3.3) and (3.5,11) for E4 and (10,10) and (4,9) for E2. For dataset X16, it consisted of 16 data points with two outliers (A and B) as illustrated in Figure 27. There were two clusters in the dataset with the true centroids at (60,150) and (140,150). Clustering was performed by setting the fuzzifier to 2 for all algorithms and computed the error using MAE as illustrated in (54) from each clustering algorithm against the nearest true centroid.

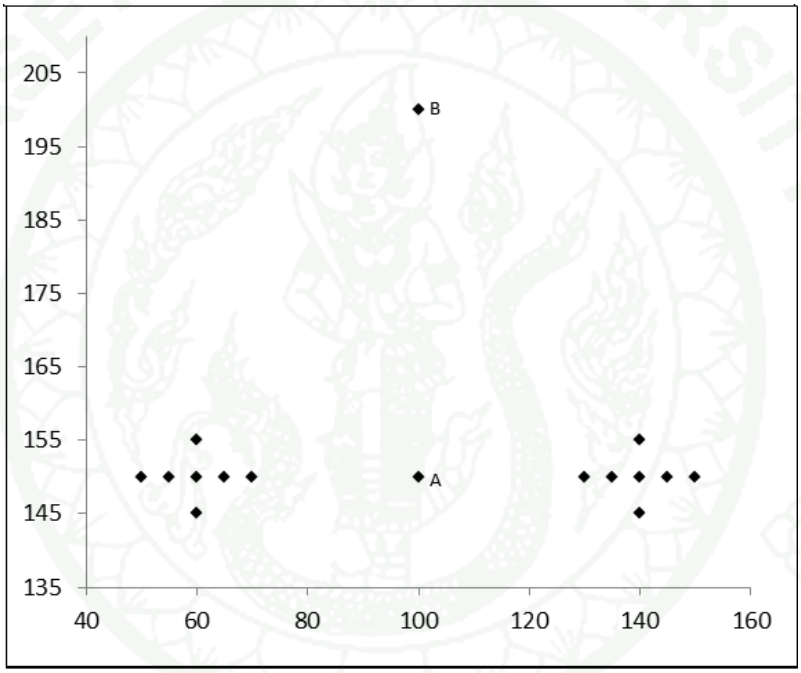

Figure 27 X16 Dataset.

The result in Table 10 shows that PXFCM produced ideal solution for X16 by generating centroids exactly the same as true centroid and excluding the two outliers. UPC excluded only the second outlier while PFCM assigned low membership value to both outliers. FCM and XFCM assigned two outliers to both clusters which were obviously result from the probability condition that sum the degree of membership across all clusters to one.

Table 10 Centroid error and degree of membership for both Outliers produced by each clustering algorithm on X16 dataset

	FCM	XFCM	UPC	PFCM	PXFCM
MAE	3.24	4.99	0.17	0.50	0.00
μ_{ij}A	0.50	0.50	0.06	0.14	0.00
μ_{ij}B	0.50	0.50	0.00	0.06	0.00

Table 11 Centroid error and degree of membership for both Outliers produced by each clustering algorithm on E2 and E4 dataset

Dataset	E2	E2N	E4	E4N
FCM1.5	2.16	3.11	**0.29**	1.31
FCM2.0	2.08	3.03	0.32	1.34
XFCM1.5	2.16	3.07	0.35	1.34
XFCM2.0	2.16	3.11	0.36	1.36
PFCM1.5	3.12*	3.12*	1.62	1.82
PFCM2.0	3.07*	3.14	1.93	2.88*
UPC1.5	3.45*	3.08	0.87	1.24
UPC2.0	3.88	3.57	1.14	0.86
PXFCM1.5	2.93	2.75	0.94	0.76
PXFCM2.0	2.53	2.89	0.71	**0.61**
PXFCM100	1.46	2.12	0.34	1.65
PXFCM5000	**1.01**	**1.87**	1.98	2.42

Another experiment on E4 and E2 dataset was setup to validate the clustering quality and stability in noisy environment. Clustering algorithms were processed by setting number of clusters to 4 and 2 respectively. The fuzzifier value was 1.5 and 2.0 which was commonly used for fuzzy clustering. For PXFCM, clustering was processed with additional fuzzifier setting at 100 and 5000. In order to test stability in noisy environment, 33 and 101 of noisy data were added randomly to the dataset E4 and E2 respectively. All clustering algorithms executed again with the

same settings. If clustering algorithms are reliable, the produced centroids should not change much comparing to the true centroids.

From the clustering result in Table 11, FCM and XFCM produced a lot more centroid errors (i.e. the changes are more than 0.5) when noisy points were added to the dataset while Possibilistic clustering (PXFCM, UPC and PFCM) produced less changes. FCM and XFCM were sensitive to noise but they yielded good clustering quality when the dataset did not contain noisy data. The actual centroids generated from possibilistic clustering algorithms were compared as displayed in Figure 28 to Figure 35. PFCM generated coincident clusters on E2 when fuzzifier set to 2.0 (Figure 34), E2N when fuzzifier set to 1.5 and E4N when fuzzifier set to 2.0 . UPC generated coincident clusters on E2N when fuzzifier set to 1.5. PXFCM did not generate coincident clusters in all cases. In addition, the clustering results produced by PXFCM were the best in most cases.

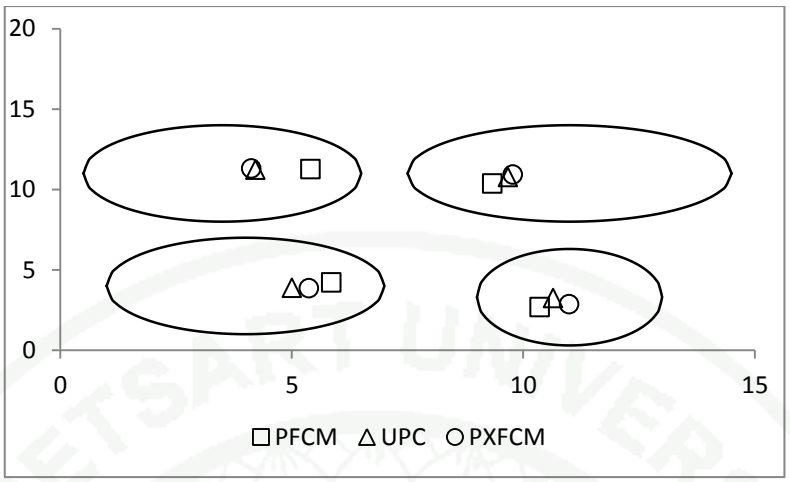

Figure 28 Centroids that generated from each Possibilistic Clustering on E4 when *m*=1.5.

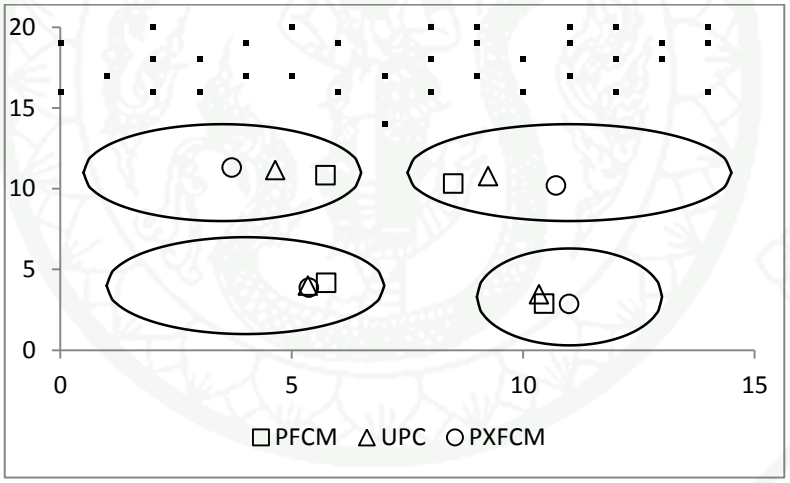

Figure 29 Centroids that generated from each Possibilistic Clustering on E4N when *m*=1.5.

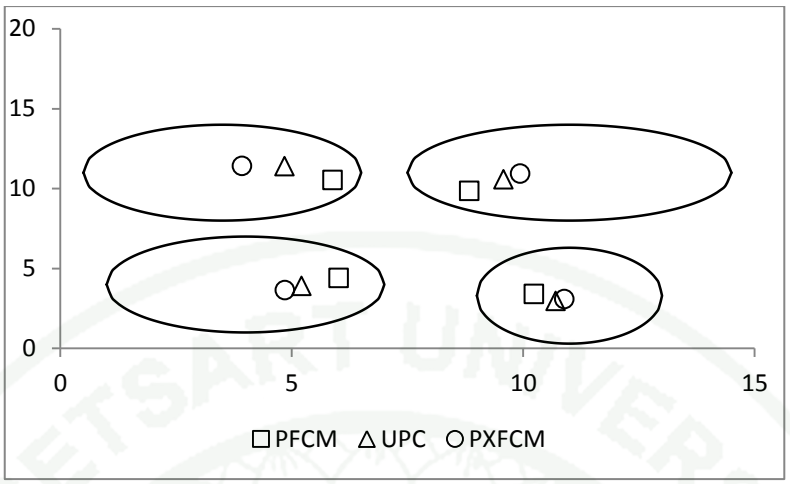

Figure 30 Centroids that generated from each Possibilistic Clustering on E4 when *m*=2.0.

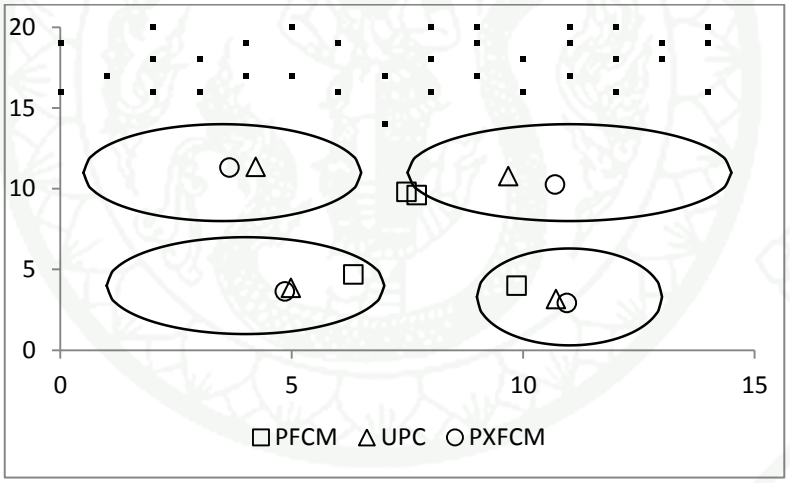

Figure 31 Centroids that generated from each Possibilistic Clustering on E4N when *m*=2.0.

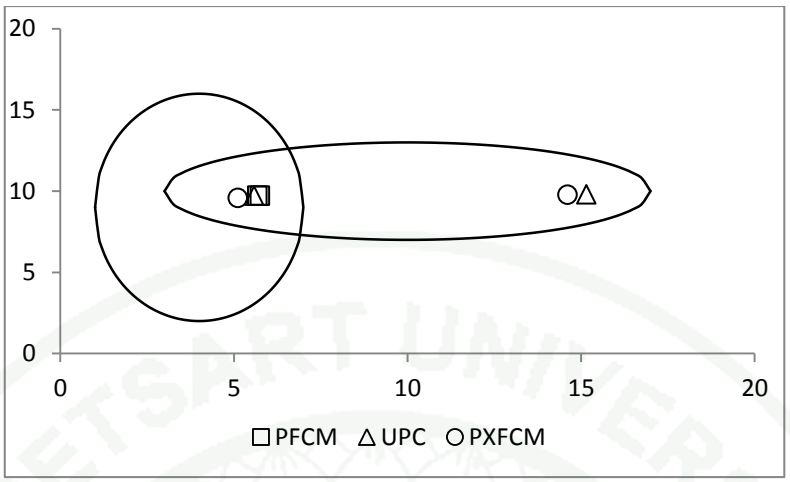

Figure 32 Centroids that generated from each Possibilistic Clustering on E2 when *m*=1.5.

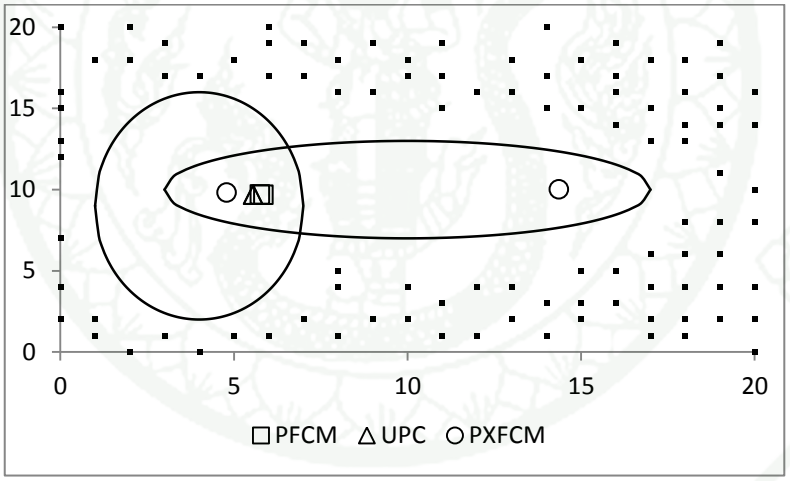

Figure 33 Centroids that generated from each Possibilistic Clustering on E2N when *m*=1.5.

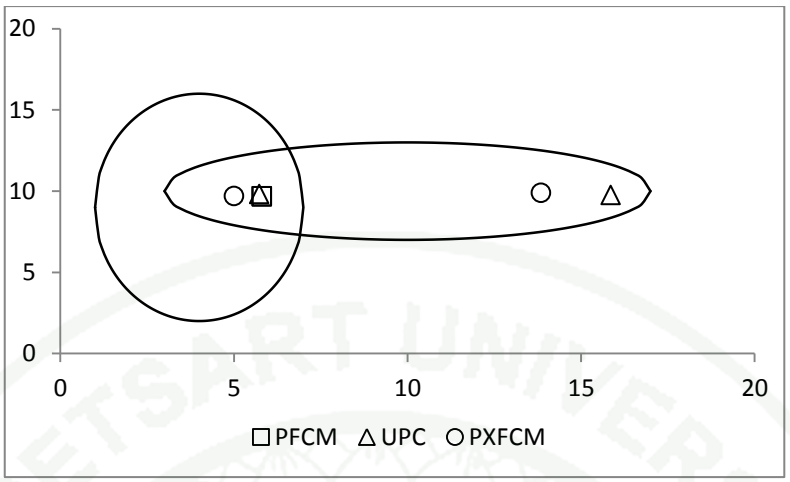

Figure 34 Centroids that generated from each Possibilistic Clustering on E2 when *m*=2.0.

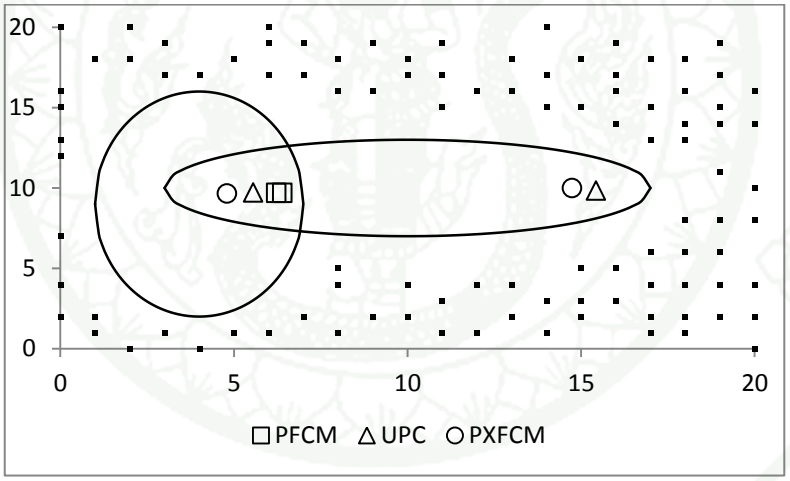

Figure 35 Centroids that generated from each Possibilistic Clustering on E2N when *m*=2.0.

For the computational time, it was obvious that FCM and XFCM processing time were better than Possibilistic clustering algorithms because these methods had less computation steps. Processing time of PXFCM was a bit longer than FCM by 12.25% and PFCM by 5.46% due to additional step to recalculate degree of membership as described in Table 12.

Table 12 Average time per iteration in seconds when clustering with fuzzy clustering algorithms on dataset E4, E4N, E2 and E2N

	FCM	XFCM	UPC	PFCM	PXFCM
E2	**3.18**	3.38	3.65	4.19	4.13
E2N	**4.36**	4.63	5.43	4.94	5.92
E4	3.92	**3.59**	10.78	3.73	3.94
E4N	5.31	5.25	5.1	4.99	**4.83**
Average	4.19	4.21	6.24	4.46	4.70

Standard datasets from UCI, such as IRIS and Wine were also used to examine the clustering quality in term of Rand Index. IRIS consists of 150 instances with 4 dimension data. There are 50 instances in each class with total of 3 classes in the dataset. Two classes are overlapped. Wine consists of 178 instances with 12 dimensions data. There are 3 classes with 59 instances in the first class, 71 instances in the second class and 48 instances in the third class. Clustering accuracy was evaluated using Rand Index as defined in (57) whereby *TP* is the number of two similar data in the same cluster. *TN* is the number of two dissimilar data in the different clusters. *FP* is the number of two dissimilar data in the same cluster. *FN* is the number of two similar data in the different clusters. Higher Rand Index indicates better clustering quality.

$$RandIndex = \frac{TP + TN}{TP + FP + FN + TN} \tag{57}$$

The result from Table 13 shows that PXFCM produced the highest Rand Index for IRIS and Wine. PFCM generated coincident clusters on Wine dataset when fuzzifier set to 1.5 and 2.0 as Rand Index was very low. UPC also generates partial coincident clusters by merging two clusters into one when fuzzifier sets to 1.5 and 2.0.

Table 13 Rand Index of each clustering algorithm when perform clustering on IRIS and Wine

Dataset	Rand Index	
	IRIS	Wine
FCM1.5	0.8620	0.7136
FCM2.0	0.8690	0.7105
XFCM1.5	0.8702	0.7105
XFCM2.0	0.8563	0.7103
PFCM1.5	0.8801	0.4160*
PFCM2.0	0.8815	0.3806*
UPC1.5	0.8537	0.5747*
UPC2.0	0.8668	0.6817*
PXFCM1.5	0.7020	0.6590
PXFCM2.0	0.8570	0.6631
PXFCM100	**0.9143**	0.6758
PXFCM5000	0.8521	**0.7142**

The result was further validated by well-known cluster validity index which measure the clustering quality in term of Compactness and Separation. Compactness indicates the variation of the data within a cluster, and separation indicates the isolation of the clusters from each other. Cluster validity index is a tool to find the optimum number of clusters. In this experiment, Xie-Beni (V_{XB}), Kwon (V_{KW}), Fukuyama-Sugeno (V_{FS}), Gath-Geva (V_{FHV}), Pakhir (V_{PMBF}) and Wu-Yang (V_{PCAES}) index were used to validate clustering results on E4.

Table 14 Optimum number of cluster for E4 obtaining from various Cluster Validity Indexes

	V_{XB}	V_{KW}	V_{FS}	V_{FHV}	V_{PMBF}	V_{PCAES}
FCM	5	2	3	5	2	2
PFCM	5	3	7	4	2	10
UPC	2	2	3	2	10	9
XFCM	5	2	3	5	2	9
PXFCM	4	2	5	4	2	2

The minimum index value from V_{XB}, V_{KW} and V_{FS} and maximum index value from V_{FHV}, V_{PMBF} and V_{PCAES} indicated the optimum number of clusters. Various clustering with various number of cluster varying from 2 to 10 were executed using m=2 and obtained the optimum number of clusters based on the index value as displayed in Table 14. Only V_{XB} and V_{FHV} returned the correct number of clusters for E4. V_{KW}, V_{PMBF} and V_{PCAES} presented monotone tendency on number of clusters while V_{XB} and V_{FHV} presented acceptable result in this experiment.

2.2 Outlier Detection

The experiments were executed following Aggrawal's approach (Aggarwal and Yu, 2001). A predictable way to test the effectiveness of an outlier detection method is to run the outlier detection method on a given dataset and test the percentage of points which belong to one of the rare classes. E2, E4 and Wisconsin Breast Cancer dataset were used to validate PXFCM performance using XOF in (41).

In these experiments, fuzzifier parameter was set to 1.5 and 2 for E2 and E4. The results were compared with LOF as illustrated in Table 15 and Table 16.

Table 15 Outlier Detection in E4N

Top Ratio (No. of Records)	Number of Outliers			
	LOF30	LOF50	PXFCM1.5	PXFCM2.0
2%(4)	**4(12.12%)**	**4(12.12%)**	**4(12.12%)**	**4(12.12%)**
7%(15)	13(39.39%)	**15(45.45%)**	**15(45.45%)**	**15(45.45%)**
10%(22)	14(42.42%)	19(57.58%)	**22(66.67%)**	**22(66.67%)**
13%(28)	14(42.42%)	22(66.67%)	**28(84.85%)**	**28(84.85%)**
16%(35)	17(51.52%)	24(72.73%)	**32(96.97%)**	**32(96.97%)**
20%(43)	18(54.55%)	26(78.79%)	**32(96.97%)**	**32(96.97%)**
30%(65)	22(66.67%)	32(96.97%)	**33(100%)**	**33(100%)**
50%(109)	31(93.94%)	**33(100%)**	**33(100%)**	**33(100%)**
70%(152)	**33(100%)**	**33(100%)**	**33(100%)**	**33(100%)**
100%(217)	**33(100%)**	**33(100%)**	**33(100%)**	**33(100%)**

The result from Table 15 indicated PXFCM1.5 and PXFCM2.0 outperformed LOF30 and LOF50 in most cases for dataset E4N. PXFCM1.5 and PXFCM2.0 performed identically. According to (38), this equation was used to determine the boundary of clusters. Since the clusters structure in dataset E4N are clearly separable hence different fuzzifier value did not have much impact to the clustering result.

Table 16 Outlier Detection in E2N

Top Ratio (No. of Records)	Number of Outliers			
	LOF30	LOF50	PXFCM1.5	PXFCM2.0
5%(15)	**15(14.9%)**	**15(14.9%)**	**15(14.9%)**	**15(14.9%)**
10%(30)	**28(27.7%)**	**28(27.7%)**	**28(27.7%)**	**28(27.7%)**
15%(45)	41(40.6%)	**45(44.6%)**	43(42.6%)	44(43.6%)
20%(61)	51(50.5%)	**58(57.4%)**	57(56.4%)	56(55.4%)
25%(76)	58(57.4%)	65(64.4%)	**68(67.3%)**	65(64.4%)
30%(91)	67(66.3%)	72(71.3%)	**75(74.3%)**	73(72.3%)
35%(106)	70(69.3%)	79(78.2%)	**83(82.2%)**	80(79.2%)
40%(121)	76(75.2%)	83(82.2%)	**86(85.1%)**	85(84.2%)
45%(136)	81(80.2%)	90(89.1%)	**91(90.1%)**	**91(90.1%)**
50%(152)	88(87.1%)	**93(92.1%)**	**93(92.1%)**	**93(92.1%)**
60%(182)	91(90.1%)	**100(99%)**	99(98%)	98(97%)
70%(212)	94(93.1%)	**101(100%)**	**101(100%)**	**101(100%)**
80%(242)	99(98%)	**101(100%)**	**101(100%)**	**101(100%)**
90%(273)	**101(100%)**	**101(100%)**	**101(100%)**	**101(100%)**
100%(303)	**101(100%)**	**101(100%)**	**101(100%)**	**101(100%)**

The result from Table 16, PXFCM1.5 and PXFCM2.0 also outperformed LOF30 and LOF50 in most cases for dataset E2N. PXFCM1.5 performed slightly better than PXFCM2.0. This could be explained with same reason as E4N. Since the clusters structure in dataset E2N are overlapped. The algorithm used (38) to determine the boundary of clusters whereby the boundary was relied on how fuzzifier was set.

For Wisconsin Breast Cancer dataset from UCI, this dataset has 699 instances with 9 attributes. Each record is labeled as benign (458 or 65.5%) or malignant (241 or 34.5%). The experimental technique was followed approach from (Ramaswamy, 2000; Willium *et al.*, 2002; He *et al.*, 2005) by removing some of malignant records to form an unbalanced distribution; the dataset has 39 (8%)

malignant records and 444 (92%) benign records. The result in Table 17 is the benchmarking result with other well-known outlier detection algorithms which provided result by He *et al.* (2005). They are LSA (He *et al.*, 2005), FFPOF (He *et al.*, 2004), FCBLOF (He *et al.*, 2003), RNN (Ramaswamy, 2000; Willium *et al.*, 2002), KNN (Ramaswamy, 2000). In these experiments, fuzzifier parameter was set to 2 for the Wisconsin Breast Cancer dataset.

Table 17 Outlier Detection in Wisconsin Breast Cancer

Top	Number of Outliers					
Ratio	PXFCM	LSA	FFPOF	FCBLOF	RNN	KNN
1%(4)	**4(10%)**	3(8%)	**4(10%)**	3(8%)	**4(10%)**	3(8%)
2%(8)	**8(20%)**	6(15%)	**8(20%)**	7(18%)	7(18%)	6(15%)
4%(16)	**16(41%)**	13(33%)	15(38%)	14(36%)	14(36%)	11(28%)
6%(24)	**24(62%)**	21(54%)	22(56%)	21(54%)	21(54%)	18(46%)
8%(32)	**30(77%)**	26(67%)	29(74%)	28(72%)	27(69%)	25(64%)
10%(40)	**34(87%)**	29(74%)	33(85%)	31(79%)	32(82%)	30(77%)
12%(48)	**38(97%)**	33(85%)	**38(97%)**	35(90%)	35(90%)	35(90%)
14%(56)	**39(100%)**	**39(100%)**	**39(100%)**	**39(100%)**	38(97%)	36(92%)
16%(64)	**39(100%)**	**39(100%)**	**39(100%)**	**39(100%)**	**39(100%)**	36(92%)
18%(72)	**39(100%)**	**39(100%)**	**39(100%)**	**39(100%)**	**39(100%)**	36(92%)
20%(80)	**39(100%)**	**39(100%)**	**39(100%)**	**39(100%)**	**39(100%)**	36(92%)
28%(112)	**39(100%)**	**39(100%)**	**39(100%)**	**39(100%)**	**39(100%)**	**39(100%)**

Table 18 Computation time in seconds for Outlier Detection

	LOF30	LOF50	PXFCM1.5	PXFCM2.0
E4N	166	177	**20**	26
E2N	108	120	65	**28**

For the Wisconsin Breast Cancer, It was clearly that among these algorithms, RNN performed the worst in most cases. The first 24 outliers that retrieved by PXFCM were the true outliers while good data points were list as outliers by other

algorithms. PXFCM outperformed other outlier detection algorithms as displayed in Table 17. In addition, PXFCM used shorter processing time comparing to LOF for all experiments as indicated in Table 18.

3. Agglomerative Fuzzy Clustering

With the AFC, this method is a generalized procedure that applicable to implement on top of any fuzzy clustering with single parameters. In this section, the experiments were setup to validate the AFC performance. Firstly, clustering results produced from AFC were benchmarked by measuring the centroid errors. The optimum centroids produced from normal clustering and clustering with adaptive AFC method were compared. All algorithms were randomly initialized the centroids. The AFC used k=10 for initial number of clusters. If AFC performed better, the clustering quality should be improved comparing to baselines that performed by normal clustering. In this experiment, FCM and XFCM were used to validate AFC performance. Secondly, the number of clusters was compared with popular CVIs. In these experiments, the number of classes was used as number of clusters. Finally, fuzzifier value was compared with estimate fuzzifier value equation (Schwammle and Jensen, 2010). Four real datasets from UCI (IRIS, Wine, Diagnostic Breast Cancer (WDBC) and Prognostic Breast Cancer (WPBC)) and 2 synthetics datasets (E2 and E4) were used in these experiments.

3.1 Clustering quality comparison

In this experiment, clustering quality was studied by comparing the centroid from each cluster algorithm against actual centroids. Better clustering algorithm should produce lower error. AFC and normal fuzzy clustering (using m=2 for FCM and XFCM) were executed on IRIS and 2 synthetics datasets (E2 and E4). Each clustering algorithm was executed 10 times per dataset and measured the centroids errors by comparing the produced centroids with the true centroids using

MAE in (54) and clustering results are in Table 19. The true centroid from IRIS obtained from Kothari and Pitts (1999). Normal clustering results were used as baseline. AFC method was effective in case of they perform better than baseline.

Table 19 Centroids errors produces from FCM and XFCM.

Dataset	FCM	XFCM	AFCM	AXFCM
IRIS	0.21	0.13	0.10	0.07
E2	11.23	12.42	2.98	12.27
E4	19.91	18.16	14.81	11.33

From Table 19, the results show that Generalized Agglomerative method that applied to both all algorithms yielded the centroid errors less than original FCM and XFCM. One reason was the number of initialization uses in Agglomerative method is larger than actual number of clusters. Consequently, the opportunity to get data from valid clusters is increased.

3.2 Number of Clusters comparison

AFCM and AXFCM were executed to obtain the number of clusters on 4 real datasets from UCI as summarized in Table 20.

Table 20 UCI dataset information for experiment

Dataset	Information
IRIS	150 instances, 3 classes, 4 attributes
Wine	178 instances, 3 classes, 13 attributes
Diagnostic Breast Cancer (WDBC)	569 instances, 2 classes, 30 attributes
Prognostic Breast Cancer (WPBC)	198 instances, 2 classes, 32 attributes

These datasets were compared to the results of FCM that validate by CVIs (Zhang *et al.*, 2008) as displayed in Table 21.

From Table 21, AFCM and AXFCM returned the number of clusters as same as actual number of clusters (k^*) and better than other well-known CVIs sometimes. In addition, time consumed by AFCM and AXFCM (see

Table **22**) was a bit lower than FCM and XFCM except E2. The lower usage time was from the number of similarity computation decreases when clusters were merged by Agglomerative method. On the other hand, FCM and XFCM needed longer time than Agglomerative method before converge. It was noted that FCM and XFCM require multiple execution of clustering process in order to obtain the right number of clusters while AFCM and AXFCM return the right number of clusters by single execution.

Table 21 Optimum Number of clusters from AFCM, AXFCM comparing to other CVIs

	IRIS	Wine	WDBC	WPBC
k^*	3	3	2	2
AFCM	3	3	2	2
AXFCM	3	3	2	2
V_{XB}	2	3	2	2
V_K	2	3	2	2
V_{FS}	5	13	12	4
V_{FHV}	3	3	2	2
V_{PMBF}	3	3	2	2
V_{PCAES}	2	3	2	2
V_W	3	3	2	2

Table 22 Execution time in seconds for Fuzzy Clustering and Agglomerative Fuzzy Clustering.

Dataset	FCM	XFCM	AFCM	AXFCM
E2	10	9	13	12
E4	20	18	16	11
IRIS	10	13	6	8
Wine	100	65	50	33
WDBC	106	70	61	66
WPBC	46	42	32	39

3.3 Fuzzifier Comparison

Fuzzifier values obtained from AFCM were compared with the estimation equation proposed by Schwammle and Jensen (2010).

Table 23 Fuzzifier value obtained from AFCM.

Dataset	E2	E4	IRIS	Wine	WDBC	WPBC
AFCM	4.0	2.9	1.8	1.3	1.1	3.2
Estimation [7]	8.5	8.7	3.2	1.3	1.1	1.1

The optimum fuzzifier obtained from AFCM on Wine and WDBC yielded the same result. However, the fuzzifier obtained from Estimation for E2 and E4 were too high. At these fuzzifier values, most membership produced from algorithm approaches *1/k* which did not seem right comparing to the structure of dataset therefore AFCM is more reliable.

Discussion

From experiments results, XFCM performed the best for both MovieLens dataset. There are several reasons why Clustering based method is better than Item based method. Basically, Item based method do not account for interaction among movies. For example, the five series of Batman in 100K MovieLens became the neighbors of some similar movies all together. This led to multiple counting of similar movies and overfitted the rating prediction. In addition, the nature of Item based method forces to find first *k* nearest neighbors which might not related to the movie. Especially, the movies had the number of useful neighbors lower than the specify *k* nearest neighbors, the ratings could be calculated from unrelated movies. In contrast to Clustering based method whereby the ratings were calculated toward cluster's centroids hence multiple counting will not happen. Also the number of related movies was controlled by the degree of membership. If the movies did not related to the clusters, the degree of membership would be low. SVD method calculates ratings based lower rank approximation thus some information can be lost. SVD method works very well if the dataset does not change. New users or new items that add to the dataset could lead SVD to recomputing the factorization sub matrices while clustering based method predict the ratings by computed similarity against cluster centroids. It does not need to update centroids if number of new users or new items is not big enough to impact the centroids.

By comparing XFCM among other clustering algorithms, XFCM performed better than FCM and FCME in term of MAE for CF domain. FCM spread the degree of membership for a data point to all clusters. This was because FCM is developed based on Polynomial function. The most common value of fuzzifier that set its value to two causes the degree of membership become linear equation. For FCME, the degree of membership is formulated in Exponential function. The results from experiments indicated the improvement of membership assignment but it did not enough to generate the high quality of rating prediction. As a result of the Exponential objective function and Logarithmic degree of membership of XFCM, it caused degree of membership behave with the three type of degree of membership as indicated in the

literature. XFCM assigned the degree of membership greater than zero only when data points and clusters were related hence only truly related data points were used to calculate the prediction.

In general, fuzzy clustering algorithms are bounded with the summation of degree of membership to one. This condition leads all data points including outliers are eventually assigned to a cluster. By relaxing this condition with Possibilistic approach, XFCM was extended to be PXFCM. From the results, PXFCM outperformed other algorithms in almost experiments for both clustering quality and outlier detections aspects. The second term of objective function of PXFCM was added to force the degree of membership to be as large as possible. PXFCM preserved the same behaviors of degree of membership as this method derived from XFCM. Possibilistic approach was another important key that broke the barrier to allow the degree of membership to represent the belonging in more meaningful. Furthermore, the normalization degree of membership procedure of PXFCM was computed indirectly to the distance of other clusters then coincident clusters can be prevented. In contrast to XFCM, PXFCM managed the negative degree of membership by turning them to be the outlier detection. PXFCM used XOF that calculated from the overall residual distance of a data point to all clusters, as associated outlier score for each data point. This method calculated based on global data points while LOF calculated score based on local neighbors. There existed some situation whereby LOF detected the good data points as outliers. For example, the first wrongly data points in E2N that detected by LOF30 was located in the low density area. If minimum points did not properly define, the data point could become outliers. With global outlier detection approach, the outliers were detected if data were located outside the boundary of dataset. In summary, different approach were used in different situation. In these experiments, the global approach using by PXFCM may probably perform better to detect outliers.

Even though clustering algorithms are effective but the information related to dataset always unknown in practice. The AFC method was developed to enable clustering to operate with minimum parameters. This method was generalized to

implement on top of fuzzy clustering algorithms with one fuzzifier parameter. From experiments, clustering quality of AFCs was better than the baselines. This was from the over specified number of cluster at initial step that increased opportunity to select good data points as initial seeds. Additionally, this method took advantage of Hierarchical clustering by obtaining optimum number of cluster during the execution. Finally, this method ensured the reasonable fuzzifier. By examining the distribution of degree of membership, the algorithm produced fuzzifier that reflected the actual information of dataset comparing to the estimated method that calculated based on number of data points and dimensions.

CONCLUSION AND RECOMMENDATION

Conclusion

Although Fuzzy Clustering had many success stories for many applications, FCM produced too much distribution for the assignment of degree of membership values. FCME, PFCM and UPC improved membership assignment but they could assign unrelated data to the cluster. In this thesis, XFCM was proposed based on the three types of members (fully belong to cluster, patially belong to cluster and not belong to cluster). Various experiments were performed on CF datasets to validate XFCM. The results showed that XFCM outperforms FCM by 5.2-9.8%, FCME by 1.0-6.1%, the Item-based method by 2.7-6.9% and SVD by 1.0-3.0% for both MovieLens datasets. Further analysis indicated that XFCM worked very well by discarding irrelevant data to update the cluster centroids. Eventually, predictions were not overwhelmed by irrelevant data.

However, Summation of XFCM's membership function of each data was bound to one. Thus, noise and outliers were eventually assigned to a cluster. XFCM was extended to relax this constraint by integrating with Possibilistic Approach. Its innovation lied in the capability to filter abnormal data and detected them as outliers. PXFCM utilized positive and negative degree of membership by turning into the outlier detection method that embeded with in the algorithm.With this functionality, it allowed PXFCM to segregate good data and abnormal data in the dataset. In addition to Possibilistic approach, the algorithm was free to assign membership degree without restriction and had better representation of degree of belonging than other fuzzy clustering. Furthermore, this method did not generate coincidence clusters and had only one parameter; hence finding the right value of parameter was easy and particle. In the outlier detection perspective, the XOF score was calculated based on the distance to the centroids on the fly during clustering process; thus XOF did not computational expensive. The quality of outlier detection was also better than other algorithms.

From the last objective, this thesis proposed a clustering automation by Agglomerative Fuzzy Clustering method. The proposed method determined the merging clusters by comparing data attributes of each centroid to prevent the merge of small clusters. This method selected the right value of fuzzifier and number of clusters parameters for fuzzy clustering. The number of clusters are usually obtained by validating the clustering result with Cluster Validity Index. Nevertheless, this method does not take fuzzifier into account. Even though, there exists a general recommendation for fuzzifier but the right value of fuzzifier for particular dataset is difficult to estimate. The proposed method was demonstrated on FCM and XFCM and validating their performance with various experiments. The results showed that Agglomerative Fuzzy Clustering obtained correct number of clusters and selected reasonable fuzzifier during the execution. This method was used to automate the algorithm and easy to operate by novice.

In summary, XFCM changes the membership distribution based on the three type of members. This approach is a promising algorithm for the allocation of data especially in CF domain. PXFCM has a greater level of the clustering tasks to the most existing clustering algorithms. When PXFCM is used for outlier detection, it holds considerable advantage in the clustering context in addition to the outlier detection. AFC is a generalized method that applicable to fuzzy clustering. This method is a clustering automation and easy to use.

Recommendation

In particular concerning of XFCM, PXFCM and AFC method, these methods could be improved by studying on different domains with larger dataset especially the outlier detection. For AFC, studying the threshold in the relation with initial seeds and property of dataset by statistical reasoning is another area of interest. Furthermore, increasing of fuzzifier could be improved with some mathematical equations. This is an area to be extended upon the future research.

LITERATURE CITED

Aggarwal, C.C. and P.S. Yu. 2001. Outlier detection for high dimensional data. **ACM SIGMOD Record** 30(2): 37-46.

Agovic, A., A. Banerjee, A.R. Ganguly and V. Protopopescu. 2007. Anomaly detection using manifold embedding and its applications in transportation corridors. **Journal Intelligent Data Analysis - Knowledge Discovery from Data Streams** 13(3):435-455.

Al-zoubil, M.B., A. Ali and A.A. Yahya. 2010. Fuzzy clustering-based approach for outlier detection, pp. 192-197. *In* **Proceeding of the 10th International Conference on Applications of Computer Engineer**. 23-25 March 2010, World Scientific and Engineering Academy and Society. Penang, Malaysia.

Ali, K. and W.V. Stam. 2004. TiVo: making show recommendations using a distributed collaborative filtering architecture, pp. 394-401. *In* **Proceeding of the 10th International Conference on Knowledge Discovery and Data Mining.** 25-25 August 2004, ACM SIGKDD. Seattle, WA, USA.

Angiulli, F., R. Ben-Eliyahu-Zohary and L. Palopoli. 2008. Outlier detection using default reasoning. **Artificial Intelligence** 172: 1837-1872.

Arima, C., K. Hakamada, M. Okamoto. and T. Hanai. 2008. Modified fuzzy gap statistic for estimating preferable number of clusters in fuzzy k-means clustering. **Journal of Bioscience and Bioengineering** 105 (3): 273-281.

Arthur, D. and S. Vassilvitskii. 2007. K-Means++: the advantages of careful seeding, pp. 1027-1035. *In* **Proceeding of the 18th Annual ACM-SIAM Symposium of Discrete Algorithms**. Society for Industrial and Applied Mathematics Philadelphia, PA, USA.

Bandyopadhyay, S. and S. Santra. 2008. A genetic approach for efficient outlier detection in projected space. **Pattern Recognition** 41(4): 1338-1349.

Banerjee, A. and R.N. Davé. 2005. The fuzzy mega-cluster: robustifying FCM by scaling down memberships. **Fuzzy Systems and Knowledge Discovery** 3613: 444-453.

Barni, M., V. Cappellini A. Mecocci. 1996. Comments on 'A Possibilistic Approach to Clustering'. **IEEE Transaction on Fuzzy Systems** 4: 393-396.

Bezdek, J.C. 1974. Cluster validity with fuzzy sets. **Journal of Cybernet** 3(3): 58-73.

Bezdek, J.C. 1974. Numerical taxonomy with fuzzy sets. **Journal of Mathematic**. **Biological** 1(1): 57-71.

Bezdek, J.C. 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, USA.

Borgelt, C. and R. Kruse. 2003. Speeding up fuzzy clustering with neural network techniques, pp: 852-856. *In* **Proceeding of the 12th IEEE International Conference on Fuzzy Systems.** 25-28 May 2003, IEEE. St. Louis, Missouri, USA.

Breunig, M.M., H. Kriegel, R.T. Ng and J. Sander. 2000. LOF: identifying density-based local outliers. **ACM SIGMOD Record** 29(2)**:** 93-104.

Cai, W., S. Chen. and D. Zhang. 2006. Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. **Pattern Recognition** 40: 825-838.

Dave, R.N. 1991. Characterization and detection of noise in clustering. **Pattern Recognition Letters** 12(11): 657-664.

Davies, D.L. and D.W. Bouldin. 1979. Cluster seperation measure. **IEEE Transaction on Pattern Analysis and Machine Intelligence** 1(2): 95-104.

Deelers, S. and S. Auwatanamongkol. 2007. Enhancing k-means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance. **World Academy of Science, Engineering and Technology** 11: 43-48.

Dunn, J.C. 1973. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. **Journal of Cybernetics** 3(3): 32-57.

Dave, R.N. 1996. Validating fuzzy partition obtained through c-shells clustering. **Pattern Recognition Letter** 17: 613-623.

Erilli, N.A., U. Yolcu, E. Eğrioğlu, Ç.H. Aladağ and Y. Öner. 2010. Determining the most proper number of clusters in fuzzy clustering by using artificial neural networks. **Expert System with Applications** 38(3): 2248-2252.

Escalante, H. J. 2005. A comparison of outlier detection algorithms for machine learning. **Programming and Computer Software** 29(4): 228-237.

Ester, M., H.-P. Kriegel, J. Sander and X. Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. pp. 226-231. *In* **Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining**. 2-4 August, 1996, ACM SIGKDD. Portland, OR, USA.

Fukuyama, Y. and M. Sugeno. 1989. A new method of choosing the number of clusters for the fuzzy c-means method, pp. 247-250. *In* **Proceeding of the 5th Fuzzy Systems Symposium.** 3 June 1989, Society for Fuzzy Sets and Systems. Kobe, Japan.

Frigui, H. and R. Krishnapuram. 1997. Clustering by competitive agglomeration. **Pattern Recognition** 30(7): 1109-1119.

Gath, I. and A.B. Geva. 1989. Unsupervised optimal fuzzy clustering. **IEEE Transaction on Pattern Analysis and Machchine. Intelligence** 11(7): 773-781.

Guo, P., Chen, C.L. and M.R. Lyu. 2002. Cluster number selection for a small set of samples using the bayesian ying-yang model. **IEEE Transaction on Neural Networks** 13(3) : 757-763.

Gustafson, D. and W. Kessel. 1979. Fuzzy clustering with a fuzzy covariance matrix, pp: 761-766. *In* **Proceeding of the 17th Conference on Symposium on Adaptive Processes**. 10-12 January 1979, IEEE CDC. San Diego, CA, USA.

Halkidi, M., M. Vazirgiannis and I. Batistakis. 2000. Quality scheme assessment in the clustering process, pp: 265-276. *In* **Proceeding of the 4th European Conference on Principles of Data Mining and Knowledge Discovery.** Springer-Verlag, London, UK.

Halkidi, M., Y. Batistakis and M. Vazirgiannis. 2002. Clustering validity checking methods: part ii. **ACM SIGMOD Record** 31(3): 19-27.

Hassan, M.A., V. Chaoji, S. Salen and M.J. Zaki. 2009. Robust partitional clustering by outlier and density insensitive seeding. **Pattern Recognition Letters** 30(11): 994-1002.

Hathaway, R.J., J.C. Bezdek and H. Yingkang. 2000. Generalized fuzzy c-means clustering strategies using LP norm distances. **IEEE Transactions On Fuzzy Systems** 8(5): 576-582.

He, Z., X. Xu and S. Deng. 2003. Discovery cluster based local outliers, **Pattern Recognition Letters** 24(9-10): 1651-1660.

He, Z., X. Xu and S. Deng. 2004. A frequent pattern discovery based method for outlier detection. **Advance in Web-Age Information Management** 3129: 726-732.

He, Z., X. Xu and S. Deng. 2005. An optimization model for outlier detection in categorical data, pp: 400-409. *In* **Proceedings of the 2005 International Conference on Advances in Intelligent Computing.** Springer-Verlag, Berlin, Heidelberg, Germany.

Herlocker, J. L., J.A. Konstan, A. Borchers and J. Riedl. 1999. An algorithmic framework for performing collaborative filtering, pp: 230-237. *In* **Proceeding of the 22nd Conference on Information Retrieval.** 15-19 August 1999, ACM SIGIR. Berkeley, USA.

Hodge, V. J. and J. Austin. 2004. A survey of outlier detection methodologies. **Artificial Intelligence Review** 22(2): 85-126.

Hou, Z., W. Qian, S. Huang, Q. Hu and W.L. Nowinski. 2007. Regularized fuzzy c-means method for brain tissue clustering. **Pattern Recognition Letters** 28(13): 1788-1794.

Hubert, L. and J. Schultz. 1976. Quadratic assignment as a general data-analysis strategy. **British Journal of Mathematic and Statistical Psychology** 29(2): 190-241.

Ichihashi, H., K. Miyagishi and K. Honda. 2002. Fuzzy c-means clustering with regularization by K-L information, pp: 924-927. *In* **Proceeding of the 10th IEEE International Conference on Fuzzy Systems**. 2-5 December 2001, IEEE. Melbourne, Australia.

Jain, A.K. 2010. Data clustering: 50 years beyond K-means. **Pattern Recognition Letters** 31(8): 651-666.

Kaur, P. 2008. DFCM: density based approach to identify outliers and to get efficient clusters in fuzzy clustering, pp: 906-909. *In* **Proceeding of the 2008 International Conference on Web Intelligence and Intelligent Agent Technology.** 9-12 December 2008, IEEE. Sydney, Australia.

Khan, S.S. and A. Ahmad. 2004. Cluster center initialization algorithm for k-means clustering. **Pattern Recognition Letters** 25(11): 1293-1302.

Kim, Y., D. Kim, D. Lee and K.H. Lee. 2004. A cluster validation index for GK cluster analysis based on relative degree of sharing. **Information Sciences - Informatics and Computer Science** 168(1-4): 225-242.

Knorr, E. and R. Ng. 1988. Algorithms for mining distance based outliers in large datasets, pp: 392-403. *In* **Proceedings of the 24th International Conference on Very Large Data Bases.** Morgan Kaufmann Publishers Inc, San Francisco, CA, USA.

Koonsanit, K. and C. Jaruskulchai. 2012. Automatic Determination of Appropriate Number of Cluster for Multispectral Image Data. **IEICE Transaction on Information and System** E95-D(5): 1256-1263.

Koren, Y. and R. Bell. 2011. Advanced in collaborative filtering, pp. 145-186. *In* F. Ricci, eds. **Recommender System Handbook.** Springer, USA.

Kothari, R. and D. Pitts. 1999. On finding the number of clusters, **Pattern Recognition Letters** 20(4): 405-416.

Krishnapuram, R. and J. Keller. 1993. A possibilistic approach to clustering. **IEEE Transaction on Fuzzy Systems** 1(2): 98-110.

Krishnapuram, R., A. Joshi, O. Nasraoui and L.Yi. 2001. Low-complexity fuzzy relational clustering algorithms for web mining. **IEEE Transactions on Fuzzy Systems** 9(4): 595-607.

Kwon, S.H. 1998. Cluster validity index for fuzzy clustering. **Electronics Letters** 34(22): 2176-2177.

Karypis, G. 2001. Evaluation of item-based top-N recommendation algorithms. *In* **Proceedings of the 12th International Conference on Information and Knowledge Management.** ACM Press, New York.

Li, M. and P.M.B. Vitanyi. 1993. **An Introduction to Kolmogorov Complexity and Its Applications**. Springer-Verlag, New York.

Li, M.J., M.K. Ng and Y. Cheung. 2008. Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters. **IEEE Transactions on Knowledge and Data Engineering** 20(11): 1519-1534.

Li, Y. and Y. Shen. 2010. An automatic fuzzy c-means algorithm for image segmentation. **Soft Computing** 14(2): 123-128.

Linden, G., B. Smith and J. York. 2003. Amazon.com recommendations: item-to-item collaborative filtering. **IEEE Internet Computing** 7(1): 76-80.

Lingras, P. and C. West. 2004. Interval set clustering of web users with rough *k*-means. **Journal of Intelligent Information Systems**. 23(1): 5-16.

Lu, B., Y. Wei and J. Li. 2009. A noise-resistant fuzzy Kohonen clustering network algorithm for color image segmentation, pp. 44-48. *In* **Proceeding of the 4th International Conference on Computer Science & Education**. 25-28 July 2009, IEEE. Nanning, China.

Lynch, M., D. Ilea, K. Robinson, O. Ghita and P.F. Whelan. 2007. Automatic seed initialization for the expectation-maximization algorithm and its application in 3D medical imaging. **Journal of Medical Engineering & Technology** 31(5): 332-340.

MacQueen, J.B. 1967. Some methods for classification and analysis of multivariate observations, pp: 281-297. *In* **Proceeding of the 5th Berkeley Symposium on Mathematic Statistics and Probability.** Univ. of Calif., USA.

Mei, J. and L. Chen. 2010. Fuzzy clustering with weighted medoids for relational Data. **Pattern Recognition** 43(5): 1964-1974.

Miyamoto, S. and M. Mukaidono. 1997. Fuzzy c - means as a regularization and maximum entropy approach, pp: 86-92. *In* **Proceeding of the 7th International on Fuzzy Systems. Associ.** IFSA World Congress, Prague.

Miyamoto, S., D. Suizu and O. Takata. 2004. Methods of fuzzy c-means and possibilistic clustering using quadratic term. **Scientiae Mathematicae Japonicae** 60(2): 217-233.

Miyamoto, S., H. Ichihashi and H. Katsuhiro. 2008. **Algorithms for Fuzzy Clustering**. Springer-Verlag Berlin Heidelberg, Chennai, India.

Mizutani, K. and S. Miyamoto. 2005. Possibilistic approach to kernel-based fuzzy c-means clustering with entropy regularization. **Modeling Decisions for Artificial Intelligence** 3558: 144-155.

Oh, C., E. Ikeda, K. Honda and H. Ichihshi. 2000. Parameter specification for fuzzy clustering by Q-learning, pp: 9-12. *In* **Proceeding of the 7th International Joint Conference on Neural Networks.** 24-27 July 2000, IEEE. Como, Italy.

Oliveira, J.V. and W. Pedrycz. 2007. **Advances in Fuzzy Clustering and its Application.** John Wiley & Sons, Ltd., Chippenham, Wiltshire, England.

Pakhira, M.K., S. Bandyopadhyay and U. Maulik. 2004. Validity index for crisp and fuzzy clusters. **Pattern Recognition** 37(3): 487-501.

Pakhira, M.K., S. Bandyopadhyay and U. Maulik. 2005. A study of some fuzzy cluster validity indices, genetic clustering and application to pixel classification. **Fuzzy Sets and Systems** 155(2): 191-214.

Pal, N.R. and J.C. Bezdek. 1995. On cluster validity for the fuzzy c-means model. **IEEE Transaction on Fuzzy Systems** 3(3): 370-9.

Pal, N. R., K. Pal and J. C. Bezdek. 1997. A Mixed C-Means Clustering Model, pp: 11-21. *In* **Proceeding of the 6th International Conference on Fuzzy Systems.** 1-5 July 1997**,** IEEE. Barcelona, Spain.

Pal, N. R., K. Pal, J. M. Keller and J. C. Bezdek. 2005. A Possibilistic Fuzzy C-Means Clustering Algorithm. **IEEE Transactions on Fuzzy Syst** 13(4): 517-530.

Ramaswamy, S., R. Rastogi and S. Kyuseok. 2000. Efficient algorithms for mining outliers from large data sets. **ACM SIGMOD Record** 29(2): 93- 104.

Redmond, S.J. and C. Heneghana. 2007. A method for initialising the K-means clustering algorithm using KD-trees. **Pattern Recognition Letters** 28(8) : 965-973.

Rendle, S. 2010. Factorization machines, pp: 995-1000. *In* **Proceeding of the 10th International Conference on Data Mining.** 13-17 December 2010, IEEE. Sydney, Australia.

Rezaee, B. 2010. A Cluster Validity Index for Fuzzy Clustering. **Fuzzy Sets and Systems** 161(23): 3014-3025.

Rousseeuw, P.J. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. **Journal of Computational and Applied Mathematics.** 20(1): 53-65.

Sarwar, B., G. Karypis, J. Konstan and J. Riedl. 2000. Application of dimensionality Reduction in Recommender System - A Case Study . *In* **Proceeding of the 2000 International Conference on WebKDD.** ACM WebKDD, Boston, USA.

Sarwar, B., G. Karypis, J. Konstan and J. Riedl. 2001. Item-based collaborative filtering recommendation algorithms, pp. 285-295. *In* **Proceeding of the 10th International World Wide Web Conference.** ACM, New York, USA.

Schwammle, V., O.N. Jensen. 2010. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. **Journal of Bioinformatics** 26(22): 1-8.

Shahi, A., R.B. Atan M.N. Sulaiman. 2009. Detecting effectiveness of outliers and noisy data on fuzzy system using FCM. **European Journal of Scientific Research** 36(4): 627-638.

Soto, J., A. Flores-Sintas and J. Palarea-Albaladejo. 2007. Improving probabilities in a fuzzy clustering partition. **Fuzzy Sets and Systems.** 159(4): 406-421.

Tang, C., S. Wang and W. Xu. 2010. New fuzzy c-means clustering model based on the data weighted approach. **Data & Knowledge Engineering** 69(9)**:** 881-900.

Tibshirani, R., G. Walther and T. Hastie. 2000. Estimating the number of clusters in a dataset via the gap statistic**. Journal of the Royal Statistical Society** 63(2): 411-423.

Treerattnapitak, K and C. Juruskulchai. 2009. Items based fuzzy c-mean clustering for collaborative filtering. **Journal of Information Technology** 5(10): 30-34.

Treerattnapitak, K and C. Juruskulchai. 2009. Entropy based fuzzy c-mean for item-based collaborative filtering, pp. 881-886. *In* **Proceeding of the 9th International Symposium on Communication and Information Technology**. 28-30 September 2009, IEEE. Incheon, Korea.

Tsao, E.C. and J. C. Bezdek. 1994. Fuzzy Kohonen clustering networks. **Pattern Recognition** 27(5): 757-764.

Turaga, D.S., M. Vlachos and O. Verscheure. 2009. On k-means cluster preservation using quantization schemes, pp:533-542. *In* **Proceeding of the 9th International Conference on Data Mining.** 6-9 December 2009, IEEE. Florida, USA.

Vozalis, M., A. Markos and K.G. Margaritis. 2009. Evaluation of standard SVD-based techniques for collaborative filtering. *In* **Proceeding of the 9th Hellenic European Research on Computer Mathematic and its Applications**. 24-26 Septempber 2009, Athens Univ. of Econ. & Busi. Athens, Greece.

Wachs, J., O. Shapira and H. Stern. 2004. A method to enhance the 'possibilistic c-means with repulsion' algorithm based on cluster validity index. **Advance in Soft Computing** 34: 77-87.

Wang, S., K.F.L. Chung, D. Zhaohong and H. Dewen. 2006. Robust fuzzy clustering neural network based on e-insensitive loss function. **Applied Soft Computing** 7(2): 577-584.

Williums G.J., R.A. Baster, H. He, S. Harkins and L. Gu. 2002. A comparative study of RNN for outlier detection in data mining, pp. 709-712. *In* **Proceeding of the 2nd International Conference on Data Mining.** 9-12 December 2002, IEEE. Maebashi City, Japan.

Wu, K. 2012. Analysis of parameter selections for fuzzy c-means. **Pattern Recognition** 45(1): 407-415.

Wu, K. and M. Yang. 2002. Alternative c-means clustering algorithms. **Pattern Recognition** 35(2002): 2267-2278.

Wu, K. and M. Yang. 2005. A cluster validity index for fuzzy clustering, **Pattern Recognition Lett**er 26(9): 1275-1291.

Xie, X.L. and G. Beni. 1991. A validity measure for fuzzy clustering. **IEEE Transaction on Pattern Analysis and machine Intelligence** 13(8): 847-847.

Xu, L. 1996. How many clusters?: a Ying-Yang machine based theory for a classical open problem in pattern recognition, pp. 1546-1551. *In* **Proceedings of the International Conference on Neural Networks.** IEEE, Washington, USA.

Xu, L. 1997. Rival penalized competitive learning, finite mixture multisets clustering. **Pattern Recognition Letters** 18(11-13): 1167-1178.

Xu., R. 2005. Survey of clustering algorithms. **IEEE Transaction on Neural Networks** 16(3): 645-678.

Xue, Z., Y. Shang and A. Feng. 2010. Semi-supervised outlier detection based on fuzzy rough C-means clustering. **Mathematics and Computers in Simulation** 80(9): 1911-1921.

Yang, M. and K. Wu. 2006. Unsupervised possibilistic clustering. **Pattern Recognition** 39(1): 5-21.

Yu, J., Q. Cheng and H. Huang. 2004. Analysis of the weighting exponent in the FCM. **IEEE Transaction on Systems, Man, and Cybernetics** 34(1): 634-639.

Zang, C. and C. Bo. 2010. Automatic Estimation the Number of Clusters in Hierarchical Data Clustering, pp. 269-274. *In* **Proceeding of the 6th International Conference on Mechatronics and Embedded Systems and Applications.** 15-17 July 2010**,** IEEE. Shandong, China.

Zhang, Y., W. Wang, X. Zhang and Y. Li. 2008. A cluster validity index for fuzzy clustering. **Information Science.** 178(4): 1205-1218.

Zhang, Y., X. Xu and Y. Ye. 2010. NSS-AKmeans: an agglomerative fuzzy k-means clustering method with automatic selection of cluster number, pp. 32-38. *In* **Proceeding of the 34th International Conference on Advanced Computer Control.** 24-29 January 2010, The American Ceramic Society. Florida, USA.

APPENDICES

Appendix A

Finding Optimization for FCM

Finding Optimization for FCM

$$J_s = \sum_{j=1}^{k}\sum_{i=1}^{N} {\mu_{ij}}^{m} \left\| x_i - v_j \right\|^2$$

From Lagrange Multiplier Minimize J_s Subject to $\sum_{j=1}^{k} \mu_{ij} = 1$

A new variable (λ) called a Lagrange multiplier is introduced, and study the Lagrange function defined by

$$L = J_s + \lambda_1 \sum_{j=1}^{k} (\mu_{1j} - 1) + \lambda_2 \sum_{j=1}^{k} (\mu_{2j} - 1) + ... + \lambda_N \sum_{j=1}^{k} (\mu_{Nj} - 1) = J_s + \sum_{i=1}^{N} \lambda_i \sum_{j=1}^{k} (\mu_{ij} - 1)$$

$$L = \sum_{j=1}^{k}\sum_{i=1}^{N} {\mu_{ij}}^{m} \left\| x_i - v_j \right\|^2 + \sum_{i=1}^{N} \lambda_j \sum_{j=1}^{k} (\mu_{ij} - 1)$$

$$L = (\mu_{11}^{m} \left\| x_1 - v_1 \right\|^2 + \mu_{12}^{m} \left\| x_1 - v_2 \right\|^2 + .. + \mu_{Nk}^{m} \left\| x_N - v_k \right\|^2) + (\lambda_1 + \lambda_2 + ... + \lambda_j)(\mu_{11} - 1 + \mu_{12} - 1 + ... + \mu_{Nk} - 1)$$

$$\frac{\partial L}{\partial \mu_{ij}} = (m{\mu_{11}}^{m-1} \left\| x_1 - v_1 \right\|^2 + m{\mu_{12}}^{m-1} \left\| x_1 - v_2 \right\|^2 + ... + m{\mu_{kN}}^{m-1} \left\| x_N - v_k \right\|^2) + (\lambda_1 + \lambda_2 + ... + \lambda_N) = 0$$

It is simplified in to general term, then the equation become

$$\frac{\partial L}{\partial \mu_{ij}} = m{\mu_{ij}}^{m-1} \left\| x_i - v_j \right\|^2 + \lambda_i = 0$$

To eliminate λ_i, the equation is rewritten to be $\mu_{ij} = \left[\frac{-\lambda_i}{m \left\| x_i - v_j \right\|^2} \right]^{\frac{1}{m-1}}$

From the condition $\sum_{i=1}^{k} \mu_{ij} = 1$ so $\sum_{i=1}^{k} \left[\frac{-\lambda_i}{m \left\| x_i - v_j \right\|^2} \right]^{\frac{1}{m-1}} = 1$ i.e.

$$\left[\frac{-\lambda_i}{m \left\| x_i - v_1 \right\|^2} \right]^{\frac{1}{m-1}} + \left[\frac{-\lambda_i}{m \left\| x_i - v_2 \right\|^2} \right]^{\frac{1}{m-1}} + ... + \left[\frac{-\lambda_i}{m \left\| x_i - v_k \right\|^2} \right]^{\frac{1}{m-1}} = 1$$

$$\left(\frac{-\lambda_i}{m} \right)^{\frac{1}{m-1}} \left(\left[\frac{1}{\left\| x_i - v_1 \right\|^2} \right]^{\frac{1}{m-1}} + \left[\frac{1}{\left\| x_i - v_2 \right\|^2} \right]^{\frac{1}{m-1}} + ... + \left[\frac{1}{\left\| x_i - v_k \right\|^2} \right]^{\frac{1}{m-1}} \right) = 1$$

Then μ_{ij} is rewritten

$$\mu_{ij} = \left[\frac{-\lambda_i}{m\left\|x_i - v_j\right\|^2} \right]^{\frac{1}{m-1}} = \frac{\left(\frac{-\lambda_i}{m}\right)^{\frac{1}{m-1}} \left[\frac{1}{\left\|x_i - v_{j_i}\right\|^2} \right]^{\frac{1}{m-1}}}{\left(\frac{-\lambda_i}{m}\right)^{\frac{1}{m-1}} \left(\left[\frac{1}{\left\|x_i - v_1\right\|^2} \right]^{\frac{1}{m-1}} + \left[\frac{1}{\left\|x_i - v_2\right\|^2} \right]^{\frac{1}{m-1}} + \ldots + \left[\frac{1}{\left\|x_i - v_k\right\|^2} \right]^{\frac{1}{m-1}} \right)}$$

$$\mu_{ij} = \frac{\left[\frac{1}{\left\|x_i - v_j\right\|^2} \right]^{\frac{1}{m-1}}}{\left[\frac{1}{\left\|x_i - v_1\right\|^2} \right]^{\frac{1}{m-1}} + \left[\frac{1}{\left\|x_i - v_2\right\|^2} \right]^{\frac{1}{m-1}} + \ldots + \left[\frac{1}{\left\|x_i - v_k\right\|^2} \right]^{\frac{1}{m-1}}} = \frac{\left[\frac{1}{\left\|x_i - v_j\right\|^2} \right]^{\frac{1}{m-1}}}{\sum_{i=1}^{k} \left[\frac{1}{\left\|x_i - v_j\right\|^2} \right]^{\frac{1}{m-1}}} = \left(\frac{\left\|x_i - v_j\right\|^2}{\sum_{i=1}^{k} \left\|x_i - v_j\right\|^2} \right)^{\frac{-1}{m-1}}$$

Finding solution for v_j by differentiate J_s

$$L = \sum_{i=1}^{N} \sum_{j=1}^{k} {\mu_{ij}}^m \left\|x_i - v_j\right\|^2 = \sum_{i=1}^{N} \sum_{j=1}^{k} {\mu_{ij}}^m ({x_i}^2 - 2x_i v_j + {v_j}^2)$$

Because the solution is for single v_j then it relaxes to

$$L = \sum_{i=1}^{N} {\mu_{ij}}^m ({x_i}^2 - 2x_i v_j + {v_j}^2)$$

From Lagrange Multiplier there is no condition for v_j so

$$\frac{\partial L}{\partial v_j} = -2 \sum_{i=1}^{N} {\mu_{ij}}^m x_i + 2v_j \sum_{i=1}^{N} {\mu_{ij}}^m = 0$$

Then it is

$$v_j = \frac{\sum_{j=1}^{N} {\mu_{ij}}^m x_i}{\sum_{j=1}^{N} {\mu_{ij}}^m}$$

Appendix B

Finding Optimization for XFCM

Finding Optimization for XFCM

$$J_s = \sum_{j=1}^{k}\sum_{i=1}^{N} \frac{m^{\mu_{ij}} - 1}{m-1} \left\| x_i - v_j \right\|^2$$

From Lagrange Multiplier Minimize J_s Subject to $\sum_{j=1}^{k} \mu_{ij} = 1$

A new variable (λ) called a Lagrange multiplier is introduced, and study the Lagrange function defined by

$$L = J_s + \lambda_1 \sum_{j=1}^{k} (\mu_{j1} - 1) + \lambda_2 \sum_{j=1}^{k} (\mu_{j2} - 1) + ... + \lambda_N \sum_{j=1}^{k} (\mu_{jN} - 1) = J_s + \sum_{i=1}^{N} \lambda_i \sum_{j=1}^{k} (\mu_{ij} - 1)$$

$$L = \sum_{j=1}^{k}\sum_{i=1}^{N} \frac{m^{\mu_{ij}} - 1}{m-1} \left\| x_j - v_i \right\|^2 + \sum_{i=1}^{N} \lambda_i \sum_{j=1}^{k} (\mu_{ij} - 1)$$

$$L = (\frac{m^{\mu_{11}} - 1}{m-1} \left\| x_1 - v_1 \right\|^2 + \frac{m^{\mu_{12}} - 1}{m-1} \left\| x_1 - v_2 \right\|^2 + .. + \frac{m^{\mu_{kN}} - 1}{m-1} \left\| x_N - v_k \right\|^2) + \lambda_1 (\mu_{11} - 1) + \lambda_2 (\mu_{21} - 1) + ... + \lambda_i (\mu_{Nk} - 1)$$

$$\frac{\partial L}{\partial \mu_{ij}} = (\frac{m^{\mu_{11}} \ln m}{m-1} \left\| x_1 - v_1 \right\|^2 + \frac{m^{\mu_{12}} \ln m}{m-1} \left\| x_1 - v_2 \right\|^2 + .. + \frac{m^{\mu_{13}} \ln m}{m-1} \left\| x_N - v_k \right\|^2) + (\lambda_1 + \lambda_2 + ... + \lambda_N) = 0$$

It is simplified in to general term, then the equation become

$$\frac{\partial L}{\partial \mu_{ij}} = \frac{m^{\mu_{ij}} \ln m}{m-1} \left\| x_i - v_j \right\|^2 + \lambda_i = 0$$

To eliminate λ_i, the equation is rewritten to be

$$m^{\mu_{ij}} = \left[\frac{-\lambda_i (m-1)}{\ln m \left\| x_i - v_j \right\|^2} \right] \text{ or } \mu_{ij} = \log_m \left[\frac{-\lambda_i (m-1)}{\ln m \left\| x_i - v_j \right\|^2} \right] = \log_m \left[\frac{(m-1)\lambda_i}{\ln m} \right] - \log_m \left[\frac{1}{\left\| x_i - v_j \right\|^2} \right]$$

From the condition $\sum_{j=1}^{k} \mu_{ij} = 1$ so $\sum_{i=1}^{k} \log_m \left[\frac{-\lambda_i (m-1)}{\ln m \left\| x_i - v_j \right\|^2} \right] = 1$ i.e.

$$\log_m\left[\frac{-\lambda_i(m-1)}{\ln m\left\|x_i-v_1\right\|^2}\right]+\log_m\left[\frac{-\lambda_i(m-1)}{\ln m\left\|x_i-v_2\right\|^2}\right]+\ldots+\log_m\left[\frac{-\lambda_i(m-1)}{\ln m\left\|x_i-v_j\right\|^2}\right]=1$$

$$k\log_m\left[\frac{(m-1)\lambda_i}{\ln m}\right]+\log_m\prod_{j=1}^{k}\left\|x_i-v_j\right\|^2=1$$

$$\log_m\left[\frac{(m-1)\lambda_j}{\ln m}\right]=\frac{1-\log_m\prod_{j=1}^{k}\left\|x_i-v_j\right\|^2}{k}$$

$$\therefore \mu_{ij}=\frac{1-\log_m\prod_{j=1}^{k}\left\|x_i-v_j\right\|^2}{k}-\log_m\left[\frac{1}{\left\|x_i-v_j\right\|^2}\right]=\frac{1-\log_m\prod_{j=1}^{k}\left\|x_i-v_j\right\|^2-k\log_m\left[\frac{1}{\left\|x_i-v_j\right\|^2}\right]}{k}$$

$$\mu_{ij}=\frac{1}{k}\left(1-\log_m\prod_{j=1}^{k}\left\|x_i-v_j\right\|^2+\log_m(\left\|x_i-v_j\right\|^2)^k\right)=\frac{1}{k}\left(1+\log_m\frac{(\left\|x_i-v_j\right\|^2)^k}{\prod_{i=1}^{k}\left\|x_i-v_j\right\|^2}\right)$$

Finding solution for v_j by differentiate J_s

$$L=\sum_{j=1}^{k}\sum_{i=1}^{N}\frac{m^{\mu_{ij}}-1}{m-1}\left\|x_i-v_j\right\|^2=\sum_{j=1}^{k}\sum_{i=1}^{N}\frac{m^{\mu_{ij}}-1}{m-1}(x_i{}^2-2x_iv_j+v_j{}^2)$$

Because the solution is for single v_j then it relaxes to

$$L=\sum_{i=1}^{N}\frac{m^{\mu_{ij}}-1}{m-1}(x_i{}^2-2x_iv_j+v_j{}^2)$$

From Lagrange Multiplier there is no condition for v_j so

$$\frac{\partial L}{\partial v_j}=-2\sum_{i=1}^{N}\frac{m^{\mu_{ij}}-1}{m-1}x_i+2v_j\sum_{i=1}^{N}\frac{m^{\mu_{ij}}-1}{m-1}=0$$

Then it is

$$v_j=\frac{\sum_{i=1}^{N}\frac{m^{\mu_{ij}}-1}{m-1}x_i}{\sum_{i=1}^{N}\frac{m^{\mu_{ij}}-1}{m-1}}=\frac{\sum_{i=1}^{N}(m^{\mu_{ij}}-1)x_i}{\sum_{i=1}^{N}(m^{\mu_{ij}}-1)}$$

Appendix C

Finding Optimization for PXFCM

Finding Optimization for PXFCM

$$J_s = \sum_{j=1}^{k}\sum_{i=1}^{N} \frac{m^{\mu_{ij}} - 1}{m-1} \left\| x_i - v_j \right\|^2 + \sum_{j=1}^{k} \lambda_j (\sum_{i=1}^{N} 1 - \mu_{ij})$$

The μ_{ij} is independent of each other. Hence, minimizing J_s with respect to μ_{ij} is equivalent to minimize following equation.

$$L = \frac{m^{\mu_{ij}} - 1}{m-1} \left\| x_i - v_j \right\|^2 + \lambda_j (1 - \mu_{ij})$$

$$\frac{\partial L}{\partial \mu_{ij}} = (\frac{m^{\mu_{ij}} \ln m}{m-1} \left\| x_i - v_j \right\|^2) - \lambda_j = 0$$

$$m^{\mu_{ij}} = \frac{\lambda_j (m-1)}{\left\| x_i - v_j \right\|^2 \ln m}$$

$$\therefore \mu_{ij} = \log_m \left[\frac{\lambda_j (m-1)}{\left\| x_i - v_j \right\|^2 \ln m} \right] = \log_m \left[\frac{\lambda_j (m-1)}{\ln m} \right] + \log_m \frac{1}{\left\| x_i - v_j \right\|^2}$$

Finding solution for v_i by differentiate J_s

$$L = \sum_{j=1}^{k}\sum_{i=1}^{N} \frac{m^{\mu_{ij}} - 1}{m-1} \left\| x_i - v_j \right\|^2 = \sum_{j=1}^{k}\sum_{i=1}^{N} \frac{m^{\mu_{ij}} - 1}{m-1} (x_i^{\,2} - 2x_i v_j + v_j^{\,2})$$

Because the solution is for single v_j then it relaxes to

$$L = \sum_{i=1}^{N} \frac{m^{\mu_{ij}} - 1}{m-1} (x_i^{\,2} - 2x_i v_j + v_j^{\,2})$$

From Lagrange Multiplier there is no condition for v_j so

$$\frac{\partial L}{\partial v_j} = -2\sum_{i=1}^{N} \frac{m^{\mu_{ij}} - 1}{m-1} x_i + 2v_j \sum_{i=1}^{N} \frac{m^{\mu_{ij}} - 1}{m-1} = 0$$

Then it is

$$v_j = \frac{\sum_{i=1}^{N} \frac{m^{\mu_{ij}} - 1}{m-1} x_i}{\sum_{i=1}^{N} \frac{m^{\mu_{ij}} - 1}{m-1}} = \frac{\sum_{i=1}^{N} (m^{\mu_{ij}} - 1) x_i}{\sum_{i=1}^{N} (m^{\mu_{ij}} - 1)}$$

Appendix D

List of Publications

NCCIT 2009: *Treerattanapitak, K and C. Jaruskulchai. 2009. Items Based Fuzzy C-Mean Clustering for Collaborative Filtering. J. Information Tech. 10 : 30-34.*

ISCIT 2009: *Treerattanapitak, K and C. Jaruskulchai. 2009. Entropy based Fuzzy C-Mean for Item-based Collaborative Filtering. In 9th International Symposium on Communication and Information Technology. pp. 881-886.*

ICONIP 2010: *Treerattanapitak, K. and C. Jaruskulchai. 2010. Membership enhancement with exponential fuzzy clustering for collaborative filtering, In Proc. of the 17th Intl. conf. on Neural info. Processing, Springer-Verlag, Berlin, Heidelberg, pp. 559-566.*

FSKD 2011: *Treerattanapitak, K. and C. Jaruskulchai. 2011. Outlier detection with Possibilistic Exponential Fuzzy Clustering, 8th Int. Conf. on Fuzzy Sys. and Know. Discovery, pp.453-457.*

JCST 2012 #1: *Treerattanapitak, K. and C. Jaruskulchai. 2012. Exponential Fuzzy C-Means for Collaborative Filtering, J. Comp Sci and Tech. 27, No.3, pp. 567-576.*

ICONIP 2012: The paper "Generalized Agglomerative Fuzzy Clustering" had been accepted and will be published in Lecture Notes in Computer Science by the year 2012.

JCST 2012 #2: The paper "Possibilistic Exponential Fuzzy Clustering" had been accepted and will be published in JCST Journal by the year 2012.

CIRRICULUM VITAE

NAME : Mr. Kiatichai Treerattanpaitak

BIRTH DATE : May 9, 1972

BIRTH PLACE : Bangkok, Thailand

EDUCATION :

YEAR	INSTITUTE	DEGREE/DIPLOMA
1994	King Mongkut Univ. (North Bangkok)	B.Sc. (Industrial Chemistry)
1999	Kasetsart Univ.	M.Eng. (Industrial Engineering)

POSITION/TITLE : SENIOR CONSULTANT

WORK PLACK : ERM-Siam Co Ltd.