Science Arena Publications
Specialty Journal of Electronic and Computer Sciences
Available online at www.sciarena.com
2017, Vol, 3 (1): 21-39

Self-Optimal Clustering by Using Fuzzy Q Learning

Saeideh soltani kamal[1*], Ehsan aminian[2]
[1*]MA of Computer Engineering , Artificial Intelligence Tendency, Islamic Azad University, Safadasht, Iran.
[2]Supervisor, Department of Computer Engineering, Faculty of Science and Research, Islamic Azad University, Tehran, Iran.
*** Corresponding Author**

Abstract: *Self-optimal clustering, in comparison with other clustering methods, includes features that, when optimizing in such environments, these characteristics should be considered. A variety of methods for clustering have already been proposed that each of these methods looks at the environment with specific approach and have optimized clustering methods by inspiration of different algorithms. In this research, we have used the Fuzzy Q learning algorithm for the first time. In a Fuzzy Q learning problem, we face with an autonomous agent that interacts with environment through trial and error, and learns to select the optimal action to reach the goal. In the Fuzzy Q learning model, the agent moves into the environment and remembers the related states and rewards. The agent tries to behave in such a way that maximizes the reward function. Since Fuzzy Q learning algorithm uses the combination of reinforcement learning and fuzzy logic, it is an appropriate option for solving this group of problems. We first define the Q learning algorithm in this thesis. And after proposing this algorithm, we will shortly investigate how to improve it by fuzzy logic, which leads to the suggestion of a fuzzy reward function to reduce the complexity of clustering, and express the efficiency of proposed algorithms by standard and appropriate tests. The results of tests indicate that the proposed method has acceptable efficiency.*

Keywords: *Q Learning, Fuzzy Logic, Clustering, Self-Optimal Clustering*

INTRODUCTION

1-1- Introduction the Structure of Thesis

Clustering is considered as a learning method without supervision, and clustering methods such as K-Means, K-Medoid, FCM, etc. are common approaches that, despite many advantages such as high speed and ease of implementation, never obtain the optimal number of clusters. A new method called Self-Optimal Clustering (SOC) is a self-optimal method, with an optimized threshold function by using an interpolation method that is effective in determining the optimal number of clusters. Determining the threshold function in the SOC through the interpolation method has had an important progress in the quality of clusters. Using this possibility, self-optimal clustering (SOC) method with a threshold function optimized by using the interpolation polynomial of Lagrange is effective in the optimal number of clusters and has better visual outcomes (Dunn J, 1973).

Our goal in this thesis is to reach a more appropriate clustering by applying a fuzzy logic in self-optimal clustering and then combining it with Q algorithm. In order to achieve a coherent and appropriate structure for performing the research, in the following of this chapter the most important principles and answering the main questions of a scientific research will be addressed.

1.2 Terms Related to Self-Optimal Clustering

Threshold Function: a positive amount that determines the proximity of data sample for the m cluster.

Lagrange: an interpolation extension method that the thresholds function is optimized in the SOC method by the Lagrange interpolation method.

Partitioning Index, Separation Index, and Dunn Index: are well known, reliable and widely used indicators used to assess the quality of clusters. These indicators are widely accepted and give very accurate and exact results.

1.3 Definition of the Problem and the Main Questions of Research

As it was stated, different methods for creating better and more precise clustering algorithms, with the use of variances and normal slices have been proposed. The self-optimal clustering (SOC) method with the threshold function optimized by using the Lagrange interpolation polynomial is more effective in the optimal number of clusters than other clustering methods. But in order to overcome the complexity of this method and to exploit the advantages of fuzzy in clustering, we use fuzzy theory in self-optimal clustering method, and then, with regard to the application of reinforcement learning algorithm in solving optimization problems to achieve better clustering, we combine the Q learning algorithm with fuzzy theory in self-optimal clustering.

In order to improve the self-optimal clustering algorithm, the following questions are raised:

・What are the methods for improving clustering in the self-optimal clustering?

・Is the use of fuzzy theory in self-optimal clustering to achieve the optimal number of clusters useful?

・How much is the impact ratio of applying the Particles Congestion Optimization Algorithm in improving self-optimal clustering by fuzzy theory?

・ Do we reach a desirable and acceptable clustering by combining the Particles Congestion Optimization Algorithm with fuzzy theory in self-optimal clustering?

4.1 Research Hypothesis

With regard to the proved capabilities of fuzzy theory in solving problems and the effective and rapid application of Q learning algorithm in solving optimization problems, our assumption is that by applying fuzzy theory in self-optimal clustering method and combining it with Q learning algorithm we achieve desirable and better clustering.

1-5. Research Objectives

The main objective of this research is to obtain the optimal number of clusters in the existing clustering methods, that in respect of achieving it, the following objectives will be pursued.

- Improving the self-optimal clustering algorithm by applying fuzzy theory,
- Presenting a better clustering algorithm by applying fuzzy theory in self-optimal clustering and combining it with the Q learning algorithm.

1.6 Research Methodology

・Studying and investigating previous proposed clustering methods

・Identifying and assessing the existing challenges and problems in the field of clustering methods

・Implementing the proposed method

・Investigating and evaluating simulations results

・Conclusion

1.7. Importance and Necessity of Performing Research

Considering that various methods have been proposed to create better and more accurate clustering algorithms by using variances, accurate analysis of problem environments and so on, hence this issue has always been considered by the researchers. So, doing researches on clustering and proposing newer techniques that can have better results than existing clustering methods is essential. Therefore, there should be techniques that are effective in finding higher quality clusters and improve clustering. In this research, by combining Q learning algorithm with fuzzy logic in self-optimal clustering for optimization, the self-optimal clustering method is introduced.

1.8 Methods and Tools of Data Analysis

To implement the proposed algorithms, MATLAB software will be used. The proposed problem and method are formulated, implemented and performed as fuzzy variables in MATLAB. The proposed method has changed the RGB color channels in the form of Low, Medium, and High fuzzy variables by fuzzy color separation technique, and it is compared with other existing clustering methods by SI, PI, and DI indices.

1.9 Methods and Tools for Collecting Data

In this thesis, resources from the internet, existing articles about the improvement of clustering methods types, books published in this field, scientific journals indexed in authoritative scientific databases, etc. have been used. Data collection tool have also been performed more as the collection of files, articles, and electronic books in the fields related to the optimization of clustering methods types.

1.10 Thesis Structure

In the second chapter of this text, we will firstly review the research overview and history about investigating clustering methods types. Then, in the third chapter, we investigate the issue of self-optimal clustering and the application of fuzzy logic in clustering and introducing Q learning algorithms for solving higher-quality clustering issues and we will examine its details. In the fourth chapter, the implementation of proposed method has been explained and the simulation results have been analyzed and investigated and compared with some existing clustering methods. Finally, in chapter five, we will conclude and investigate the advantages and disadvantages of algorithm and we will have suggestions for the continuation of work.

2.1 Introduction

Clusters are the collections of objects that are similar to each other, that is, objects belonging to a cluster must be different from the objects in other clusters. The process of clustering objects can be fuzzy or hard. In the hard clustering algorithm, each element is considered as a cluster, while in the fuzzy clustering method, each member is assigned a membership grade depending on the degree of connection to several clusters (Bezdek JC, 1980; Bezdek JC, 1973). Fuzzy C-mean clustering is one of the well-known methods of clustering that can obtain good results in terms of cluster validity (Honda K et al, 2010; Ren Y et al, 2012). In Modified Mountain Clustering (MMC), when a point of a probable cluster is determined, the potentials of other points reduce. However, considering this limitation, we want to delete specific points, which include probable cluster centers too (Verma NK et al, 2007). In Improved Mountain Clustering (IMC), after determining the center of first probable cluster, a cluster is formed around this center and will be removed from the rest of data points, thus, it can maintain the potential of the remaining data points. This method gives us better results in terms of cluster validity and time complexity (Verma NK et al, 2009; Verma NK et al, 2011; Felzenszwalb PF et al, 1998). The threshold function defined in the IMC, which is subjectively determined always leaves a domain open for better optimization of the threshold function (Shi J et al, 2000). Using this possibility, self-optimal clustering (SOC) with threshold function, optimized by using Lagrange interpolation polynomial, is effective in optimal number of clusters, and has better visual results (Dunn J, 1973)

2.2 Self-Optimal Clustering Structure

Self-Optimal Clustering (SOC) is a self-optimal method with optimized threshold function, by using an interpolation method that is effective in determining the optimal number of clusters. Determining the threshold function in the SOC through the interpolation method has had a significant improvement in the quality of clusters.

The stages of SOC algorithm are as follows

1. Normalizing the data for each dimension of problem environment, so that the data points are surrounded by one unit of few cubic. The J sample of data in the full environment of X is defined as follows:

(1.a) $$X^j = \{X_1^j, X_2^j, \ldots, X_D^j\}$$

(1.b) $$\bar{X}^j = \frac{x^j - (X)_{min}}{(X)_{max} - (X)_{min}} \quad ; \forall_j = 1,2,\ldots,n$$

(2) $$(X)_{min} = \quad min_{j=1}^{n} x_1^j, min_{j=1}^{n} x_2^j, \ldots, min_{j=1}^{n} x_D^j$$

(3) $$(X)_{max} = \quad max_{j=1}^{n} x_1^j, max_{j=1}^{n} x_2^j, \ldots, max_{j=1}^{n} x_D^j$$

(D is the total number of the dimensions of problem environment and n is the total number of data points in the data set.)

2. Determining the threshold amount (δm) which is positive and defines the approximate point of data for the m cluster.

The δm is a subjective phrase of βm for the m cluster. The beta is calculated through the interpolation method, which is described in the next section of algorithm. When it gives the m cluster for the first time, beta is considered as a single amount.

(4)

$$.\beta_m \delta_m = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{min x^j}{\sum_{i=1}^{D} x_1^j} \right)$$

3. Determining the probable amount of data point r in the m cluster by using the Mountain function.

(5) $$P_r^m = \sum_{j=1}^{n} exp\left[- \left(\frac{d^2(\bar{x}^r, \bar{x}^j)}{\delta_m^2} \right) \right]$$

4. Choosing the data point related to the highest amount in $P_m^1, P_m^2, \ldots, P_m^n$ as the m cluster and the Cm center

(6)

$$\bar{c}_m = \bar{X}^* \Leftarrow P_m^* = max_{r=1}^{n}(P_m^r)$$

5. Assigning the data points in the data set to the m cluster whose Euclidean distance from the center of m cluster is less than the threshold amount of δm

(7) $d^2(\bar{X}^r, \bar{c}_m) \leq \delta_m$; $\forall_r = 1,2, \dots, n$

6. Removing all data points from the data set assigned to the m cluster.

7. Repeating stages 2 to 6 to reduce the data set and forming consecutive clusters, equal to the optimal number of M clusters.

8. Distributing the remaining data points among formed clusters based on their Euclidean distance to the center of cluster.

As it was stated, SOC is the new type of IMC method, which is very similar to IMC up to this stage.

9. Calculating the total profile amount through the GSI (Global Silhouette Index) for obtained clusters by using:

$$s(i) = \frac{b(i) - \mathrm{a(i)}}{\max\{a(i), b(i)\}}$$

$$GSI = \frac{1}{\mathrm{M}} \sum_{m=1}^{M} S_m$$

The close amount of GSI to a unit indicates a better cluster formation, this is realized when the amounts of Sm profile for m = 1,2, ..., M move towards a constant amount.

Suppose that t of every cluster has been composed, that a threshold amount is δt and the profile amount is St.

10. The interpolation formula is used in this way that for the whole clusters of M, M to a pair of amount exists as $(\delta_2, S_2), (\delta_1, S_1) \dots . (\delta_M, S_M)$.The interpolation polynomial is as the following figure:

(8)$\delta_t = l_m(\delta_t)$

11. Replacing St equal with one unit, obtained in polynomial.

(9)

$$l_m(\delta_t) = \prod_{k=1,k!=m}^{M} \frac{(\delta_t - \delta_k)}{(\delta_m - \delta_k)}$$

$$= \frac{(\delta_t - \delta_1)}{(\delta_m - \delta_1)} \cdots \frac{(\delta_t - \delta_{(m-1)})}{(\delta_m - \delta_{(m-1)})} \frac{(\delta_t - \delta_{(m+1)})}{(\delta_m - \delta_{(m+1)})} \cdots \frac{(\delta_t - \delta_M)}{(\delta_m - \delta_M)}$$

$$S_1 . \frac{(\delta_t, \delta_2)}{(\delta_1, \delta_2)} \frac{(\delta_t, \delta_3)}{(\delta_1, \delta_3)} \cdots \frac{(\delta_t, \delta_M)}{(\delta_1, \delta_M)}$$

$$+ S_2 . \frac{(\delta_t, \delta_1)}{(\delta_2, \delta_1)} \frac{(\delta_t, \delta_3)}{(\delta_2, \delta_3)} \cdots \frac{(\delta_t, \delta_M)}{(\delta_2, \delta_M)} + \cdots$$

$$+ S_5 . \frac{(\delta_t, \delta_1)}{(\delta_5, \delta_1)} \cdots \frac{(\delta_t, \delta_4)}{(\delta_5, \delta_4)} \frac{(\delta_t, \delta_6)}{(\delta_5, \delta_6)} \cdots \frac{(\delta_t, \delta_M)}{(\delta_5, \delta_M)}$$

$$+ \cdots + S_M . \frac{(\delta_t, \delta_1)}{(\delta_M, \delta_1)} \frac{(\delta_t, \delta_2)}{(\delta_M, \delta_2)} \cdots \frac{(\delta_t, \delta_{M-1})}{(\delta_M, \delta_{M-1})}$$

12. Using the above formula to find the δt roots. (In the above formula all amounts except δt are specified.)

13. Replacing the obtained roots and returning one by one of them to the polynomial in the above formula. Selecting this root for St, this is the closest to the unit. Selected root (η), the threshold amount of δt, is related to the maximum amount of St.

14. Dividing the root by δm to obtain beta amounts which is in the formula of second stage (determining the threshold amount) and calculating the new δm.

(11)

$$\beta_m = \frac{\eta}{\delta_m} \ ; \ \forall_m = 1,2,\dots,M$$

15. Repeating stages 2 to 14 until the δm is merged.

2.3 Self-Optimal Clustering Characteristics

2.3.1 Beta Calculation

The SOC method has used a number of valid, widely used, and authentic indicators for measuring the quality of clusters: Partitioning Index, Separation Index, and Dunn Index, which are all valid indicators based on inter-cluster and intra-cluster distances. For better cluster quality, the inter-cluster distances should be as high as possible, and the intra-cluster distances between the data points that make clusters should be as low as possible. The variation in δ_m causes changes in cluster quality, and inter-cluster and intra-cluster distances and also changes the amounts of various validation indices. Therefore, by changing δ_m, validation indices can be improved in a higher quality clustering to optimize them. Among all the methods of interpolation, the Lagrange interpolation method is chosen, since the distribution of δ_m in color classification will not be uniform and monotonous in all cases. Therefore, the selected root η is calculated from the polynomial to obtain the amount of threshold function δ_t. The amount η is divided by δ_m to obtain the amount of β_m. We can improve the quality of clusters by multiplying δ_m to the β_m optimizing factor. Using the new amounts of β_m, the stages of algorithm are repeated until the threshold amount of $\delta_m$is obtained. Based on the extensive experiments and the accurate observations of results in all cases, the number of repetitions of particular stages in the algorithm is selected as ten times to ensure optimal threshold amount.

2.3.2 Cluster Quality Measurement

Various valid scales have been used as clusters quality scale. These valid scales are widely accepted and yield very correct and accurate results. These scales are explained clearly below (Dunn J, 1973 and Verma NK et al, 2011).

2.3.2.1 Partitioning Index

PI: Partitioning Index is the ratio of total compressions and the decomposition of clusters.

$$PI = \sum_{m=1}^{M} \frac{\sum_{j=1}^{n} (\mu_{jm})^2 \|\bar{X}^j - \bar{c}_m\|^2}{N_m \sum_{k=1}^{M} \|\bar{c}_k - \bar{c}_m\|^2}$$

Here $\bar{C}_m$ is the m cluster center on m, N_m is fuzzy cardinality; that is, the sum (μ_{jm}), μ_{jm} is the member of j data point in the m cluster.

Lower amount of PI represents a better partitioning.

2.5.2.2 Separation Index

SI: Separation Index uses a minimum distance separation for valid clustering, that lower amount of SI represents a better partitioning.

$$SI = \frac{\sum_{m=1}^{M} \sum_{j=1}^{n} (\mu_{jm})^2 \|\bar{X}^j - \bar{c}_m\|^2}{n. min_{k,m} \|\bar{c}_k - \bar{c}_m\|^2}$$

2.3.2.3 Dunn Index

DI: Dunn Index can specify a set of well-compacted and isolated clusters. For each partitioning V ↔ X : X1 ∪ X2 ∪…..X_m, that $X_m$represents the m cluster of each partitioning, the valid Dunn index, that is DI, is defined as follows.

$$DI = min_{1\le m\le M}. \left\{ min_{1\le k\le M, k!=m} \left\{ \frac{d(X_m, X_k)}{max_{1\le m\le M}\{\Delta(X_m)\}} \right\} \right\}$$

Here, $d(X_m, X_k)$ is the average distance of inter-cluster centers that determines the distance between clusters of X_m and X_k. The $\Delta(X_k)$ which shows the full diameter of inter-cluster distance is the X_m cluster. The main purpose of this measurement is to maximize inter-cluster distances and to minimize intra-cluster distances. Therefore, the large amount of DI is related to the good clusters, so the number of clusters that maximize DI can be considered as the optimal number of M clusters.

2.3.3. Optimal Threshold Function Required

The threshold function in the SOC is systematically optimized in order to obtain the best clustering with this method by using the complementary method. The quality of cluster is better when the amount of DI is higher and the amount of SI and PI is lower. With the optimized threshold function, the SOC algorithm yields better results. Improving the quality of clustering is done well by relative increase of the amount of DI and significant reduction of the amounts of SI and PI.

2.4 Modeling Self-Optimal Clustering

The purpose of this section is the computational evaluation of features in the calculation of the initial threshold function and the threshold function in successive repetitions, and also demonstrating the optimality of SOC algorithm as well as simulations. In mathematical computations, an unlimited set of numbers are merged absolutely if the sum of absolute amounts is limited. More precisely, the real or complex set of $\sum_{n=0}^{\infty} a_n$ is merged perfectly, provided that $\sum_{n=0}^{\infty}|a_n| = L$ is real or complex for some L numbers. Also, the unfinished integral of a function $\int_0^{\infty} f(a)dx$ is perfectly merged, provided that the absolute integral amount is the limited integral (sub-integral function), which is as: $\int_0^{\infty} f(a)dx = L$. However, in mathematical computations, there are collections or integrals that can be merged, although they are not merged perfectly. Therefore, in a merged set, we call each set that has not been merged perfectly, conditional merge. This topic is represented as a set of $\sum_{n=0}^{\infty} a_n$ and is conditionally merged if $\lim_{m\to\infty} \sum_{n=0}^{\infty} a_n$ is created and the number is limited, i.e. it does not evaluate ∞ or -∞, but evaluates $\sum_{n=0}^{\infty}|a_n| = \infty$. We have the initial threshold function shown in formula 4. Using formulas 4 and 11 we will have:

(19)

$$\delta_m = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{minx^j}{\sum_{i=1}^{D} x_1^j} \right) . \left(\frac{\eta}{\delta_m} \right)$$

It should be mentioned that the δ_m stated on the right side of formula (19) shows that the threshold function has obtained its amount by calculating in the previous reciprocal of the equation which helps to evaluate the

δ_m of the left side of formula (19). We removed the lower sign of m and insert I into the formula (20), which indicates the formation of a cluster in its own I repetition. So we have:
(20)

$$\delta_I = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{minx^j}{\sum_{i=1}^{D} x_1^j} \right)_I . (\beta_{I-1})$$

In replacing by using formula (11) we have:
(21)

$$\delta_I = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{minx^j}{\sum_{i=1}^{D} x_1^j} \right)_I . \left(\frac{\eta_{I-1}}{\delta_{I-1}}\right)$$

By expanding the formula (21) more using the formula (4):

$$\boldsymbol{\delta_I} = \left(\frac{1}{2n} \sum_{j=1}^{n} \frac{minx^j}{\sum_{i=1}^{D} x_1^j} \right)_I . \left(\frac{\boldsymbol{\eta_{I-1}}}{\left(\frac{1}{2n} \sum_{j=1}^{n} \frac{minx^j}{\sum_{i=1}^{D} x_1^j} \right)_{I-1} .(\boldsymbol{\beta_{I-2}})} \right)$$

Consequently, in the formula (23), for each point sample x, we will have a spectrum of probable amounts of x (R, G, B) as (0, 0, 0) to (255,255,255).
This phrase has a minimum amount of 0 and a maximum of 0.3333. This phrase has been obtained by MATLAB simulating, that attempts to use Nelder-Mead's simple algorithm to convert a vector x, which is a minimizing of mathematical function.

(23)

$$(\frac{1}{2_n} \sum_{j=1}^{n} (\frac{minx^j}{\sum_{i=1}^{D} x_1^j})) = [0,0.3333]$$

β_0 in the first repetition of formula (4) is considered as a unit.

(24)

$$\left(\frac{1}{2_n}\sum_{j=1}^{n}\left(\frac{minx^j}{\sum_{i=1}^{D} x_1^j}\right)\right)_I = [0,0.1666]$$

That leads to the formation of $\delta_1$as a constant and is fixed with a determined amount in a spectrum. Therefore, from the formulas (23) and (24):
(25)

$$\delta_I = (const.)_I . f(\eta_{I-1})$$

The initial threshold function with a beta threshold factor has a constant amount of "1" that is not merged, because it has a factor of $\frac{1}{n}$ in its function. Since $\lim_{N\to\infty} \sum_{n=1}^{N}(\frac{1}{n})$ is an infinite set, it affects merging the initial threshold function. The initial threshold function is merged with the beta multi-purpose factor. Therefore, the calculated threshold function in each of the repetitions is conditionally merged. However, based on the conditional merge of the threshold function, this function can adapt itself automatically and generate the most optimal threshold function, and then create clusters with better quality.

2.5 Performance of Clustering Methods

The results analysis of clustering in different images shows that the SOC quality is better than other clustering methods; in different images analyzed in the comparison process, the amounts of total SOC profile compared to other clustering methods are very good in most cases. Valid indicators that are used for analysis support the high superiority of SOC. Famous clustering methods like K-Means, K-Mediod, FCM, EM, MMC, and IMC have been compared with the SOC method for the results of partitioning. While performing this work, we find out that groups' centers have widely been separated in FCM. One of the disadvantages of FCM is that it is sensitive to the primary partitioning and may remain as the minimum of function criterion. Cluster credit amounts for some of the clusters retrieved by K-Means are more than those retrieved by the FCM, but these cases are very few. EM clustering leads to the creation of unsatisfactory amounts in terms of valid indicators. K-Medoid results are very close to the K-Means results, but never show better performance than other clustering methods as FCM, which in most cases is predominate. IMC-2 results are much better than IMC-1 in most cases, but the advanced type of SOC not only improves IMC-1 results, but also is predominant to the better quality of clusters and valid indicators amounts, compared with all clustering methods such as IMC-2. Therefore, the SOC method creates the best clusters among all clustering methods.

2.6 Conclusion

The SOC method is an optimized and advanced IMC method. The performance of several popular clustering methods, such as K-Means, K-Mediod, FCM, EM, MMC, and IMC, has been compared with the SOC method for classification results. While performing this work we found out that EM cannot produce several clusters and it is not as competitive as other clustering methods. IMC-2 often produces good results, but in some cases leads to poorly clustering compared to IMC-1. In these cases, SOC is highly accepted with its acceptable results. The results of tests based on valid indicators indicate the priority and superiority of SOC method.

3.1 Introduction

The use of fuzzy control systems is regarded as another common method for setting the parameter of various algorithms, including the Q learning algorithm. In this kind of control systems, the fuzzy parameter is based on a set of fuzzy rules that is introduced by the designer of algorithm (Gamio JC et al, 2004). This set of fuzzy rules, according to the environmental conditions, the search process, and the mode, is able to select the amounts of the required parameters of algorithm appropriately. The use of fuzzy system eliminates the need to adjust the parameters of algorithm and also provides the possibility that the amounts of these parameters change and adapt to the current situation over the time of search. This is the adaptation over time that provides the possibility of improving the results relative to the state of constant amounts (Comaniciu D et al, 1997).

One of the problems most of the proposed algorithms for clustering are involved in is to identify extreme points in the environment, that sometimes these amounts are overlapping with each other, and since they deal with parameters and numerical amounts, this problem can partly be solved by fuzzy logic. In this chapter, a new algorithm called Fuzzy Q learning algorithm has been proposed to improve self-optimal clustering. The proposed algorithm is obtained by combining the proposed Q learning algorithm in the previous chapter and the fuzzy logic (Comaniciu D, 2002).

Fuzzy logic is among the multi-value logics, and relies on the theory of fuzzy sets. The fuzzy sets themselves are obtained from the generalization and expansion of definite sets in a natural way. The knowledge required for many of the issues under study appears in two distinct ways: objective knowledge such as models, equations and mathematical functions that are pre-set and are used to solve common problems of physics, chemistry or engineering. Personal knowledge such as the information that is partly describable and linguistic expression, but it is not usually possible to quantify them with the help of traditional mathematics. This kind of knowledge is called tacit knowledge or implicit knowledge (Liu J et al, 1994; Deng Y et al, 2001).

Since both types of knowledge are needed in practice, fuzzy logic tries to coordinate them with each other in a regular, logical, and mathematical manner. The fuzzy system designed for this algorithm has been composed of various parts that the most important of them are:

- Getting information from the environment
- Making the input information Fuzzy
- Defining fuzzy sets
- Defining fuzzy rules
- Establishing relationships between observations and fuzzy sets
- Evaluating each case for all fuzzy rules

・Combining information obtained from the rules

・Making the results de-fuzzy.

3.2 Reinforcement Learning Algorithms

The purpose of reinforcement learning is to teach how to map states into actions in order to maximize a numerical reward signal. The learner is not told what to do, like most models of machine learning, but the learner must discover, by performing actions, which action results in the highest reward. In most interesting and challenging situations, actions not only affect immediate rewards, but also the next state and in this way all subsequent rewards are affected. These two characteristics, and its test and error search nature, and the delayed rewards are two distinctive and important features of reinforcement learning (Gdalyahu Y et al, 2001).

One of the challenges occurring in reinforcement learning, and not in other types of learning is the compromise between exploration and extraction. In order to obtain a high reward, a learning agent should prefer the actions he has done in the past and found out that they are effective in generating rewards. But in order to discover such actions, he must choose actions that have not been selected previously. The agent must exploit what he has learned so far in order to obtain the reward, but he also has to explore the best actions in the future (Lucchese L et al, 2001) The problem is that none of the exploration or exploitation can be pursued in a specific way without failure. The agent must try different actions and consider those that seem to be better more and more.

The Q learning algorithm is an expanded mode of the Value Iteration algorithm, which is also used for nondeterministic problems. The Q Learning is a kind of non-model reinforcement learning that operates on the basis of random dynamic programming. In Q learning, instead of doing a mapping from states to the amounts of states, the mapping from the state/action pair to the values that are called Q-value is performed.

3.2.1 Elements of Reinforcement Learning

In addition to the factor and the environment, four main sub-elements of a reinforcement learning system can be specified: a policy, a reward function, a value function, and a model of environment, which is an optional element (Bezdek JC, 1973).

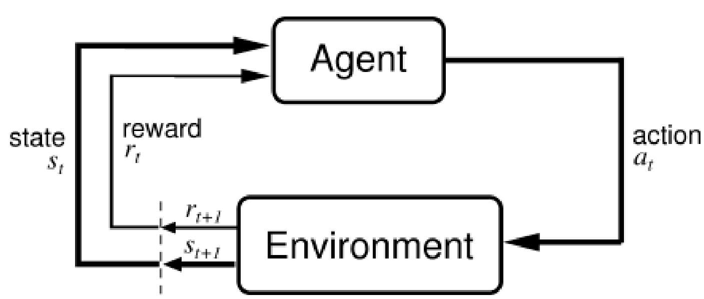


Figure 1: Interaction of Environment and Agent in Reinforcement Learning (Bezdek JC, 1980)

3.2.2 Policy

A policy specifies the behavior of learner agent at any time. Inaccurately, a policy is a mapping of the observed states of environment to actions that must be performed when the agent is in those states. In some states, policy may be a simple function or a selective table while in other states may include massive calculations, such as a search process. Policy is the core of a reinforcement learning agent, because it is solely sufficient for specifying the behavior. In general, policy may be coincidental.

3.2.2 Reward Function

A reward function defines the goal in a reinforcement learning problem. Inaccurately, it maps each observed state (or state/action pair) of environment to a reward signal state; a reward specifies the inherent degree of desirability of that state. The exclusive objective of reinforcement learning agent is to maximize the overall reward that the agent receives in the long time. In a biological system, rewards can be expressed as the same as pains and pleasures.

3.2.4 Value Function

While a reward function specifies what is good in an immediate state, a value function specifies what is good in the long time. Inaccurately, the value of a state is the total amount of reward that an agent is expected to accumulate from the beginning of that state in the future. While the rewards specify the optimal degree of the immediate inherent of environment states, the values specify the long-term desirability of states, after calculating the states that are likely to be followed, and the rewards in which the states are available. For example, a state may always result in a low immediate reward, while having a high value, because it has been followed with states that result in high rewards, or vice versa. Biologically, rewards are like pleasure, if they are high, and are like pain, if they are low, while values, in accordance with a more conscious judgment, is about the ratio of goodness or badness of that state.

3.3 Fuzzy Q Learning Algorithm Process for Self-Optimal Clustering

As you observed in the previous chapter, the proposed algorithm consisted of several important stages; these stages are parameter valuing, reward matrices formation, state and action, and a search table. In the upcoming algorithm, we intend to offer a fuzzy model for optimization in the self-optimal clustering by the aid of Q learning algorithm. Initially, we state the stages of proposed algorithm stages step-by-step, then we present the pseudo-code of proposed algorithm, and in the last section we perform numerous experiments on the proposed algorithm.

If we want to explain the performance of proposed algorithm in a paragraph, it is in this way that we make the reward process fuzzy, that is, instead of giving the s, a pair numeric reward, we use fuzzy values for giving reward. And in order to control the reward process properly, we create a rule base and, regarding the e-Greedy principle, we ensure convergence in the rules; in the way that each rule defined in the system has membership degree of P, and for them we compute the e-Greedy. In this way a reward is defined for each state, and after evaluating the fuzzy rules, we make the results de-fuzzy. In the table below, you can observe the most important parameters of proposed algorithm.

Table 3.1: Fuzzy Q Learning Algorithm Parameters (FQLO)

Parameter	Description
FQ(s,a)	Q matrix holding value function

P	Membership degree
ɵ	e-Greedy parameter
R	Rewarding matrix
V_{π}	Optimal policy
DEQ(s,a)	Adaptive matrix on the environment search
S	Defined states
A	Defined actions for movement

3.3.1 Making the Input Data Fuzzy (Rewards as Fuzzy)

We should use language variables to make the input data fuzzy. Language variables refer to the variables that accepted values for them are words and sentences of human or machine languages, instead of numbers. As in the mathematical calculations, numerical variables are used in fuzzy logic language (spoken or non-numeric) variables. Language variables are expressed on the basis of language (spoken) values which are in the phrase set (words/terms). Language phrases are attributes for language variables. For example: similarity in clustering can be defined with a few variables, such as very low, low, medium, high, and very high, and for each of these variables, we define a membership degree.

Membership degree of xμ indicates the ratio of membership of the x element to the fuzzy set. If the membership degree of an element of the set is equal to zero, that member is completely out of the set, and if the membership degree of a member is equal to one, that member is completely in the set. Now, if the membership degree of a member is between zero and one, this number represents the gradual membership degree. This feature in the proposed algorithm causes the complexities to decrease when finding optimums. As you observe in the figure below, the specified points can be categorized more easily by receiving membership degree, while in the previous method we had to define a separate state for each point.

3.3.2 Membership Function

A membership function (MF) is a curve that shows how each point of the input environment is written as a membership value (membership degree) between 0 and 1. The input environment is often referred as the world of discussion. One typical example for fuzzy sets is the set of people's height. In this case, the world of debate is that of all possible heights from 3 to 9 feet, and the word "Height" shows a curve that defines a degree for anyone's height. If the height set of people is given with a definite boundary of the classical set, we can say that all states with a height of more than 6 feet are formally tall. But making such a difference is completely meaningless. Such a classification may bring this notion to the mind that our collection is a set of all real numbers greater than 6. But when we want to talk about real people, there is no reason to consider a person tall and the other short, when the difference of their height is in their hair size.

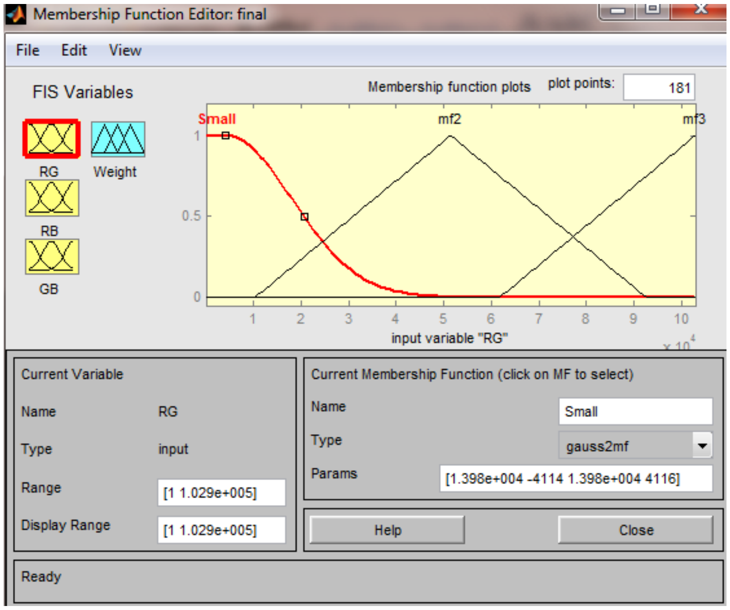


Figure 3-2: Setting Membership Functions in MATLAB

The only condition that a membership function should really satisfy is that its amount should be between 0 and 1. The function itself can be any kind of arbitrary function. The shape of this function can be in respect of simplicity, convenience, and efficiency to attract our satisfaction. A fuzzy set is a developed set of a classical set. If X is the debate world and we show its members by x, then the fuzzy set A of X defines an ordered pair. Following these descriptions, the most important features of this method can be mentioned as the following cases:

- Using the Q table dynamically and defining the reward function,
- Using Mamdani Fuzzy Inference System,
- Fuzzy look at the environment where the agent is facing with,
- Experimenting on the subject of moving peaks.

The basis of work is that we first define the search environment in fuzzy sense, and obtain if - then functions for the state and action. The fuzzy rules are defined as follows. In the rule below, s represents the states, and the function B indicates the degree of membership for each state.

3.1 $$R_i\text{: If } s_1 \text{ is } B_{1i} \text{ and ... } s_N \text{ is } B_{Ni} \text{ then } a \text{ is } a_i \text{ and } q \text{ is } q_i$$

The function B is computed as follows.

3.2 $$\beta_i(\mathbf{s}) = \mu_{B_{1i}}(s_1) \cdot \mu_{B_{2i}}(s_2) \cdot \ldots \cdot \mu_{B_{Ni}}(s_N)$$

After making the environment Fuzzy and obtaining the Fuzzy rules of if- then, and their relationships with state environment, we must obtain new Fuzzy relations for Q learning equations.

3.3

$$Q(\mathbf{s}, A(\mathbf{s})) = \frac{\sum_{i=1}^{I} \beta_i(\mathbf{s}) \cdot q_i}{\sum_{i=1}^{I} \beta_i(\mathbf{s})}$$

4.3

$$A(\mathbf{s}) = \sum_{i=1}^{I} \bar{\beta}_i(\mathbf{s}) \cdot a_i$$

5.3

$$q_i \leftarrow q_i + \alpha \bar{\beta}_i(\mathbf{s}) \Delta Q(\mathbf{s}, a)$$

The above equations show the results from top to bottom respectively: 1. Results for table Q, 2. Reward function, and 3. Updated amounts are inside the table. Then, to sum up, it can be said that this algorithm makes the environment (states) fuzzy by using the fuzzy inference system, and with the fuzzy look at the environment reduces its complexity and produces satisfactory results.

3.2.2 Ensuring the Convergence with the e-Greedy Property

This factor determines the quality of rules for us. The result of low numerical relationship is between zero and one. The closer it is to one, the quality of law is better.

6.3

$$\tilde{Q}_t(X_t, U(X_t)) = \frac{\sum_{Ri \in A(X_t)} \alpha_{R_i}(X_t)\, w_t^i(U_t^i)}{\sum_{Ri \in A(X_t)} \alpha_{R_i}(X_t)}$$

This property has been described in the implementation section, that you can refer to the attachments for further reading.

3.2.4 Creating a Fuzzy Inference System (TS)

Among the advantage of fuzzy logic is that, it connects the input environment to the output environment with a mapping, and the primary principles for doing so, as it was stated in the preceding chapters, are a series of if- then phrases, that we called them the rule. All rules are valued in parallel form and the order of rules does not matter at all. Rules are important alone, because they all refer to the variables and describe the variables. Before we begin to construct a system that interprets the rules, we must define all the phrases that describe the type of system usage, as well as the related attributes, which describe the type of usage. To do this, we use the MATLAB fuzzy toolbox. This toolbox removes most of our needs regarding the fuzzy inference systems.

3.2.5 Fuzzy Logic Toolbox

The fuzzy logic toolbox contains 11 prefabricated membership functions. These 11 functions have been composed of various important functions: Fragmentation - Linear Functions, Gaussian Distribution Function, Circular

Curve, Two and Three Degree Polynomial Functions. By convention, all functions have the letters mf at the end of their names.

The simplest membership functions are formed of right lines. The simplest function is a triangular membership function and the function name is the trimf. The points that make up the triangle are not more than three points. Trapezium membership function, trapmf, has a smooth upper surface and is in fact it is a triangular curve that its head is cut off.

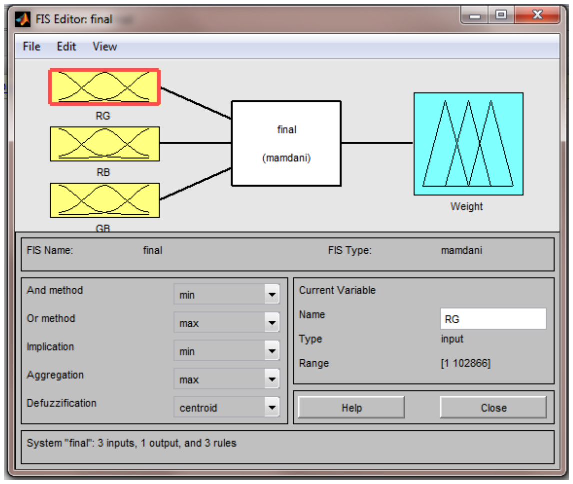
Two membership functions have also been made according to the Gaussian distribution curve: a simple curve and a two-sided combination of two separate Gaussian curves. These two functions are gaussmf and gauss2mf.

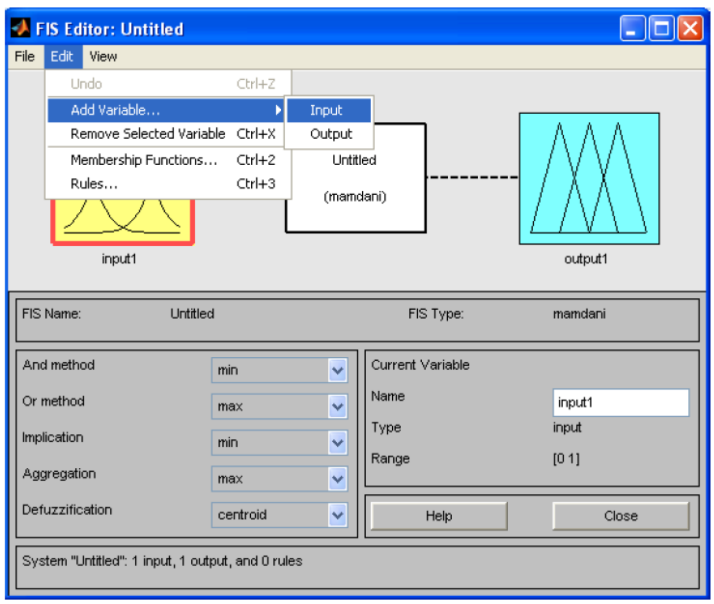
The generalized bell membership function is specified to three parameters, and its function's name is gbellmf. The bell membership function has one more parameter than the Gaussian membership function. So, if we consider the free parameter for the function, it can reconcile itself with non-fuzzy sets. Due to its low smoothness and complexity, Gaussian and bell functions are among the popular methods for specifying fuzzy sets. Among the advantages of these two functions are that they are smooth and also do not become zero at any point.

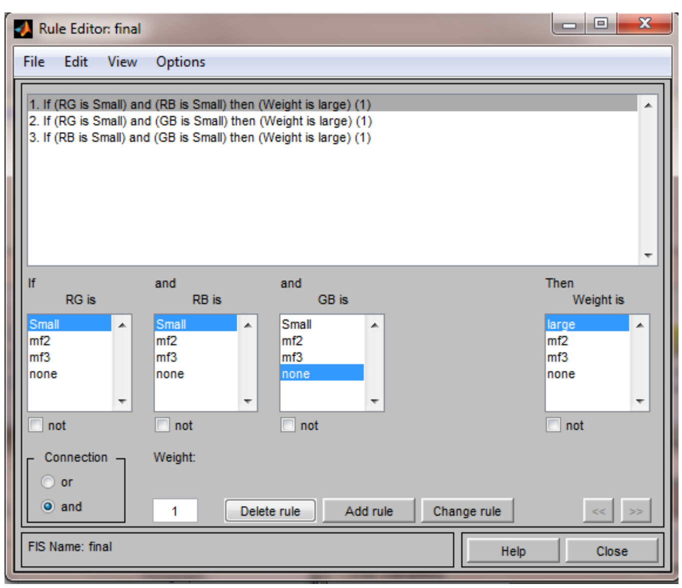
Although the Gaussian and Bell membership functions are easily obtained, they are not able to specify non-symmetric membership functions. These asymmetric functions play an important role in specific applications. In the following, curve membership functions are discussed that can be open and free from the left or right. Asymmetric and closed membership functions (i.e. not open from left or right) can be obtained by the combination of two curve functions, so in addition to the main function of sigmf, we have also the difference of two curve functions, dsigmf and the multiplication result of two psigmf curve functions.

Functions built on polynomial basis are used to construct different membership functions that are in the fuzzy logic toolbox. Three functions are related to these types of functions, curves of S, Z and Pi. It should be mentioned that naming this type of functions is based on their shape. The zmf function is an asymmetric polynomial function open from the left side. The smf function is an open function from the right side that has mirror symmetry compared with the previous curve, and the pimf function has a zero value at its two ends and a bump in its middle.

When you want your desired membership function, you have many choices. Of course, if you see this series of functions a limiter for yourself, the fuzzy logic toolbox allows you to create your favorite membership function. On the other hand, if you think that the list of prefabricated functions is confusing, keep in mind that during your work, you can probably get the desired result by using two functions, for example, two triangular and trapezoidal functions. If possible, provided that someone is willing to review the functions, he will find a lot of functions.


FIS Editor: final
File Edit View
RG
RB
GB
final
(mamdani)
Weight
FIS Name: final
FIS Type: mamdani
And method min
Or method max
Implication min
Aggregation max
Defuzzification centroid
Current Variable
Name RG
Type input
Range [1 102866]
Help
Close
System "final": 3 inputs, 1 output, and 3 rules


FIS Editor: Untitled
File Edit View
Undo Ctrl+Z
Add Variable...
Input
Output
Remove Selected Variable Ctrl+X
Membership Functions... Ctrl+2
Rules... Ctrl+3
Untitled
(mamdani)
input1
output1
FIS Name: Untitled
FIS Type: mamdani
And method min
Or method max
Implication min
Aggregation max
Defuzzification centroid
Current Variable
Name input1
Type input
Range [0 1]
Help
Close
System "Untitled": 1 input, 1 output, and 0 rules


Rule Editor: final
File Edit View Options
1. If (RG is Small) and (RB is Small) then (Weight is large) (1)
2. If (RG is Small) and (GB is Small) then (Weight is large) (1)
3. If (RB is Small) and (GB is Small) then (Weight is large) (1)
If RG is
and RB is
and GB is
Then Weight is
Small
mf2
mf3
none
large
not
Connection
or
and
Weight:
1
Delete rule
Add rule
Change rule
FIS Name: final
Help
Close

Comparing the evaluation parameters:

```
GSI =

   0.518378808670429


PI =

   0.071986045108979


SI =

   0.014582617230901


ADI =

   0.566868317722139

Elapsed time is 1325.703883 seconds.
```

	Number of Features			
F	1	3	7	20
0.1	18.88	7.78	7.89	7.32
0.2	18.98	7.63	7.67	7.25
0.3	19.02	7.66	7.36	7.1
0.4	17.67	7.49	7.37	7.03
0.5	17.88	7.25	7.27	6.87
0.6	17.66	7.37	7.22	6.74
0.7	16.65	6.99	6.96	6.77
0.8	16.18	6.85	6.75	6.61
0.9	16.12	6.84	6.8	6.63
1.0	16.33	6.92	7.08	6.81
Lowest	16.12	6.84	6.75	6.61
Highest	19.02	7.78	7.89	7.32
Optimal Amount	0.9	0.9	0.8	0.8

Testing the effect of the number of features at different frequencies

Conclusion

The SOC method is an optimized and advanced IMC method. The performance of several popular clustering methods, such as K-Means, K-Mediod, FCM, EM, MMC, and IMC, has been compared with the SOC method for classification results. While performing this work we found out that EM cannot produce several clusters and it is not as competitive as other clustering methods (Lucchese L et al, 1999). IMC-2 often produces good results, but in some cases leads to poorly clustering compared to IMC-1. In these cases, SOC is highly accepted with its acceptable results. The results of tests based on valid indicators indicate the priority and superiority of SOC method (Bruce J et al, 2000). One of the problems most of the proposed algorithms for clustering are involved in is to identify extreme points in the environment, that sometimes these amounts are overlapping with each other, and since they deal with parameters and numerical amounts, this problem can partly be solved by fuzzy logic (Abonyi J et al, 2002). In this chapter, a new algorithm called Fuzzy Q learning algorithm has been proposed to improve self-optimal clustering (Dempster AP et al, 1977; Yager RR et al, 1994). The proposed algorithm is obtained by combining the proposed Q learning algorithm in the previous chapter and the fuzzy logic (Comaniciu D, 2002).

Therefore, based on the results obtained and the suggested methods, evaluation and comparison in this paper, we reach the following results:

Optimization of dynamic environments using the Q learning algorithm

Creation of new fuzzy reward function algorithm to reduce knowledge dependency in a dynamic environment

Defining a new FQ algorithm to optimize dynamic environments

Reducing knowledge dependence (enhancing self-reliance) in the time of searching

Creation of a fuzzy inference system to overcome continuous space problems

Defining a new Q neurofuzzy learning algorithm to optimize dynamic environments

References:

[1] J. Dunn, A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, Taylor & Francis, 1973.

[2] J.C. Bezdek, Fuzzy mathematics in pattern classification (Ph.D. thesis), Appl.

Math., Cornell Univ., Ithaca, NY, 1973.

[3] J.C. Bezdek, A convergence theorem for the fuzzy isodata clustering algorithms, IEEE Trans. Pat. Anal, Mach. Intell. (1) (1980) 1–8.

[4] K. Honda, A. Notsu, H. Ichihashi, Fuzzy PCA-guided robust K-means clustering,

IEEE Trans. Fuzzy Syst. 18 (1) (2010) 67–79.

[5] Y. Ren, X. Liu, W. Liu, Dbcamm: a novel density based clustering algorithm via using the mahalanobis metric, Appl. Soft Comp. 12 (5) (2012) 1542–1554.

[6] N. K. Verma and M. Hanmandlu, "Color segmentation via improved mountain clustering technique," Int. J. Image Graph., vol. 7, no. 2, pp. 407–426, Apr. 2007.

[7] N. K. Verma, P. Gupta, P. Agarwal, M. Hanmandlu, S. Vasikarla, and Y. Cui, "Medical image segmentation using improved mountain clustering approach," in Proc. 6th Int. Conf. ITNG, Las Vegas, NV, USA, 2009, pp. 1307–1312.

[8] N. K. Verma, A. Roy, and S. Gupta, "Color segmentation using improved mountain clustering technique version-2," in Proc. 2nd IEEE Int. Conf. Intell. Human Comput. Interact., Allahabad, India, 2011, pp. 536–542.

[9] P. F. Felzenszwalb and D. P. Huttenlocher, "Image segmentation using local variation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Santa Barbara, CA, USA, 1998, pp. 98–104.

[10] J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 8, pp. 888–905, Aug. 2000.

[11] J. C. Gamio, S. J. Belongie, and S. Majumdar, "Normalized cuts in 3-D for spinal MRI segmentation," IEEE Trans. Med. Imag., vol. 23, no. 1, pp. 36–44, Jan. 2004.

[12] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: Color image segmentation," in Proc. Comput. Vision Pattern Recognit., 1997, pp. 750–755.

[13] D. Comaniciu, "Image segmentation using clustering with saddle point detection," in Proc. IEEE Int. Conf. Image Process., Rochester, NY, USA, 2002, pp. III-297–III-300.

[14] J. Liu and Y. H. Yang, "Multiresolution colour image segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 16, no. 7, pp. 689–700, Jul. 1994.

[15] Y. Deng and B. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 8, pp. 800–810, Aug. 2001.

[16] Y. Gdalyahu, D. Weinshall, and M. Werman, "Self organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization," IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 10, pp. 1053–1074, Oct. 2001.

[17] L. Lucchese and S. K. Mitra, "Color segmentation through independent anisotropic diffusion of complex chromaticity and lightness," in Proc. Int. Conf. Image Process., 2001, pp. 746–749.

[18] L. Lucchese and S. K. Mitra, "Unsupervised low-frequency driven segmentation of color images," in Proc. Int. Conf. Image Process., Kobe, Japan, 1999, pp. 240–244.

[19] J. Bruce, T. Balch, and M. Veloso, "Fast and inexpensive color image segmentation for interactive robots," in Proc. IEEE Int. Conf. Intell. Robots Syst., 2000, pp. 2061–2066.

[20] J. Abonyi, R. Babuska, and F. Szeifert, "Modified Gath-Geva clustering for identification of Takagi-Sugeno fuzzy models," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 32, no. 5, pp. 612–621, Oct. 2002.

[21] A. P. Dempster, N.M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," J. Roy. Stat. Soc., Ser. B, vol. 39, no. 1, pp. 1–38, 1977.

[22] R. R. Yager and D. P. Filev, "Approximate clustering via the mountain method," IEEE Trans. Syst., Man, Cybern., vol. 24, no. 8, pp. 1279–1284, Aug. 1994.