Opinion Dynamics and the Effect of Time-varying Opinions: A Simulation Study

Kai Yan

A thesis presented for the degree of

Master of Applied Science in Engineering

uOttawa

L'Université canadienne
Canada's university

Department of Graduate Studies in Mechanical Engineering,

University of Ottawa, Canada

August, 2015

Abstract

Opinion dynamics is extensively used in studying large-scale social, economical, political and natural phenomena that involve many interacting agents. It also can be used to model the evolution of teams of autonomous vehicles operating in a coordinated fashion with civilian and military applications, when arbitration among individual goals needs to be negotiated. Recently, research was conducted on how opinion dynamics can be the core of collective decision-making mechanisms for swarm robotics. Opinion dynamics with a time varying opinion space, which is the set of all possible opinions an agent may have, is a relatively recent research topic.

In this work, the Deffuant-Weisbuch model (DW model), which allows to model opinion dynamics in shrinking opinion spaces, was applied. In simulating this class of systems and in extracting information from them it is crucial to establish reliable algorithms and criteria for counting the numbers of clusters, as this ultimately affects the determination of the steady state of the system. A method was applied to combine Fuzzy c-means clustering and subtractive clustering to check convergence of the system and avoid negative influence of outliers. Different scenarios are simulated to study the influence of characteristic parameters on the formation of opinions, which is quantified by the formation of clusters in the opinion space. Additionally, we simulate the scenario of a two dimensional opinion space in which one side shrinks, and evaluate how the rate of

shrinking influences the steady state opinion space. This is a simplified model to gain some insight on the effect of extreme changes of opinions in multi-dimensional opinion space.

Acknowledgment

I would like to express my appreciations to my supervisor Dr. Davide Spinello for his patience, support, inspiration, direction and suggestions. With his help, I overcome countless difficulties in the research and finally, find the right direction of my thesis. I also thank my friends Yuetling Fung, Jin Bai, and Yu Hu for their encouragement and advices. Finally, I would like to thank my parents for their love, care, patient and ongoing support.

Table of Contents

Table of Contents V
List of Figures VII
List of Tables X
1 Introduction 1
1.1 Motivation 1
1.2 Research Objective 2
1.3 Thesis Outline 3
2 Background 4
2.1 Opinion Dynamics 4
2.1.1 Opinion Dynamics Models 5
2.1.2 Opinion Dynamics Applications 7
2.2 Data Clustering 9
2.2.1 What is data clustering? 10
2.2.2 Clustering methods and applications 11
2.2.3 Clustering algorithms 13
2.3 Summary 22
3 The Deffuant-Weisbuch Model and Assessment of Different Clustering Algorithms 24
3.1 The Deffuant-Weisbuch Model 24
3.2 Determine the Proper Clustering Algorithms 26
3.2.1 K-means 27
3.2.2 Gaussian Mixture Models 29
3.2.3 Fuzzy C-means Clustering 30
3.2.4 Subtractive Clustering 32
3.3 Summary 34
4 Study on clustering and opinion dynamics related to DW model 36
4.1 Estimate the number of clusters 37
4.1.1 Steady State and Verification of the Exit Condition 39
4.1.2 Outliers and Confirmation of the Number of Clusters 41
4.2 Simulation with DW model 47
4.2.1 Initial Condition: Randomly Distributed data 48
4.2.2 Initial Condition: Regularly Distributed Data 51
4.2.3 Initial Condition: Different Number of Data Points 54
4.2.4 Study on Time Varying Opinion Space 56
5 Conclusion 66
5.1 Summary 66
5.2 Research Contributions 67
5.3 Future Work 67
References 69
Appendix A 74
Convergence Time Analysis 74
Appendix B 81

MATLAB: Opinion Dynamics Code.. 81

List of Figures

Fig. 2.1 Diversity of clusters. The seven clusters in (a) differ in shape, size and density. Although these clusters are apparent to a data analyst, none of the available clustering algorithms can detect all these clusters [30]. 10

Fig. 2.2 K-means separates data into Voronoi-cells [48]. 16

Fig. 2.3 Examples on Gaussian mixture model [45]. 17

Fig.3.1 Initial condition of data points for testing clustering techniques 27

Fig.3.2 Partitioning data by using kmeans function. a) when k equals 2; b) when k equals 3; c) when k equals 4 28

Fig.3.3 Partitioning data by using GMM method. a) when k equals 2; b) when k equals 3; c) when k equals 4 30

Fig.3.4 Partitioning data by using FCM. a) when k equals 2; b) when k equals 3; c) when k equals 4 32

Fig.3.5 Partitioning data by using Subtractive Clustering. a) when radii equals 0.1; b) when radii equals 0.3; c) when radii equals 0.5; d) when radii equals 0.7 33

Fig.4.1 Algorithm procedure for ascertaining the number of clusters 38

Fig.4.2 Result after 40,000 iterations running. a) Distribution of data points. b) shows the belongings of each data point. 40

Fig.4.3 Result after 9,000 iteration steps. 40

Fig.4.4 An overlapped case when 0.1 is the confidence bound; (a) shows the result in a steady state (b) shows the number of data points 42

Fig.4.5 An example of silhouette method. (a) is the original data. (b) shows the silhouette criterion values for each number of clusters tested. (c) creates a grouped scatter plot to visually examine the suggested clusters. 45

Fig.4.6 An example of silhouette method in an overlapping data case. (a) is an original data (b) shows the silhouette criterion values for each number of clusters tested..... 46

Fig.4.7 An example of Subtractive Clustering in an overlapped data case 47

Fig.4.8 Initial condition with randomly distributed data 48

Fig.4.9 Simulation with the DW model in two-dimensional cases. (a) ε equals 0.1. (b) ε equals 0.11. (c) ε equals 0.12. (d) ε equals 0.13. (e) ε equals 0.14. (f) ε equals 0.15. (g) ε equals 0.2. (h) ε equals 0.25. (i) ε equals 0.3. (j) ε equals 0.35.(k) ε equals 0.4. 50

Fig.4.10 Average number of clusters versus ε 51

Fig.4.11 Initially regularly distributed data 52

Fig.4.12 Average number of clusters in varying initial conditions. a) Initially regularly distributed data points; b) compare to initially randomly distributed data points. 53

Fig.4.13 Initial condition of randomly distributed data: (a) the number of data points is 1000 ;(b) the number of data points is 800; (c) the number of data points is 400; (d) the number of data points is 200..... 54

Fig.4.14 Average number of clusters in varying number of data points. 55

Fig.4.15 A unit box with 1600 randomly distributed points 57

Fig.4.16 Simulation ten times at rate of 10,000 iterations when(a) ε equals 0.1. (b) ε equals 0.15.(c) ε equals 0.2.(d) ε equals 0.25.(e) ε equals 0.3.(f) ε equals 0.35.(g) ε equals 0.4. (The horizontal axis is the number of runs, and the vertical axis is the opinion space)59

Fig.4.17 Bifurcation diagram..... 60

Fig.4.18 Simulation ten times at rate of 1,000 iterations when (a) ε equals 0.1; (b) ε equals 0.15; (c) ε equals 0.2; (d) ε equals 0.2.5; (e) ε equals 0.3; (f) ε equals 0.35. (The horizontal axis is the number of runs, and the vertical axis is the opinion space)..... 61

Fig.4.19 Simulation ten times at rate of 5,000 iterations: when a) ε equals 0.1; b) ε equals 0.15. 62

Fig.4.20 Average number of clusters at different rate of iteration under different bound of confidence..... 63

Fig.4.21 Comparison between time-varying and static opinion space 64

Fig.A1 Convergence iterations when ε equals 0.1. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters 74

Fig.A2 Convergence iterations when ε equals 0.12. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters..... 75

Fig.A3 Convergence iterations when ε equals 0.15. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters..... 75

Fig.A4 Convergence iterations when ε equals 0.2. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters 75

Fig.A5 Convergence iterations when ε equals 0.25. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters..... 76

Fig.A6 Convergence iterations when ε equals 0.3. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters 76

Fig.A7 Convergence iterations when ε equals 0.35. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters..... 76

Fig.A8 Convergence iterations when ε equals 0.4. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters 77

Fig.A9 The 2nd time simulation when ε equals 0.2. (a) final convergence state. (b) the relation between simulation iterations and number of clusters 78

Fig.A10 The 3rd simulation time when ε equals 0.2. (a) final convergence state. (b) the relation between simulation iterations and number of clusters 78

Fig.A11 The 4th simulation time when ε equals 0.2. (a) final convergence state. (b) the relation between simulation iterations and number of clusters 78

Fig.A12 Simulation the 5th time when ε equals 0.2. (a) final convergence state. (b) the relation between simulation iterations and number of clusters 79

Fig.A13 The 6th simulation time when ε equals 0.2. (a) final convergence state. (b) the relation between simulation iterations and number of clusters 79

List of Tables

Table 4.1 Average number of clusters under the initially randomly-distributed data condition 51

Table 4.2 Average number of clusters under the initially regularly-distributed data condition (ten runs) 52

Table 4.3 Average number of clusters with different data points from 1000 to 200 (ten runs) 55

Table 4.4 The average number of clusters related to the bound of confidence at rate of 10,000 iterations per step. 59

Table 4.5 The average number of clusters related to the bound of at the rate of 1,000 iterations per step. 62

Table 4.6 The average number of clusters related to the bound of confidence in static opinion space. 63

Table 4.7 The average number of clusters related to the bound of confidence when the convergence happened in every step of shrinking 65

Table A1 Convergence iterations under different bound of confidence 77

Table A2 Six times of simulation when ε equals 0.2 79

1 Introduction

1.1 Motivation

Opinion dynamics has been widely used to study large-scale social, economical, political and natural phenomena that involve many interacting agents, primarily due to its application in finding a common agreement about issues of containing a group of agents. “Examples for discussing groups are parliaments, a commission of experts or citizens in a participation process. The opinion issues in parliaments can be tax rates or items of the budget plan, in commissions of experts predictions about macroeconomic factors and for citizens the willingness to pay taxes or the commitment to a constitution” [1]. Recently, there is a great interest in deploying opinion dynamics in robotics. A team of autonomous vehicles operating in a coordinated fashion is one of the research fields that can involve opinion dynamics. Potential applications for multivehicle systems include combat, surveillance, and reconnaissance systems, which need various cooperative control capabilities including formation control, rendezvous, flocking, and cooperative search[2]. One of the practical challenges for cooperative control of multiple autonomous vehicles is arbitration among individual goals. This problem in terms of negotiation of individuals recently attracted researchers’ attention by using opinion dynamics. For example, in some cases, opinion dynamics models can be the core of collective decision-making mechanisms for swarm robotics [3].

Nearly all relevant publications in terms of opinion dynamics focus on a time invariant

opinion space, where the opinion space is the set of all possible opinions an agent may have. Opinion dynamics with a time-varying opinion space is a relatively recent research topic. Applications of the theoretical framework of opinion dynamics when the states are defined in shrinking opinion spaces include the modeling of a group of drones landing (from three-dimension to two-dimension), or a team of robots going across a narrow bridge (from two-dimension to one-dimension). This thesis is dedicated to the study of a problem involving a time-varying opinion space, to understand if and how this feature may influence the steady-state opinion distribution.

1.2 Research Objective

The main objective of this thesis is to evaluate the influence of time-varying opinion space on the steady-state opinion distribution. More specifically, this study was undertaken in order to understand how the rate of shrinking of opinion space produces an effect in opinion interaction, and to gain some insight on the effect of the extremization of opinions in multi-dimensional opinion spaces. Two-dimensional cases with time invariant opinion spaces are also studied as compared to the cases of shrinking opinion spaces. For the sake of getting the process and result of opinion distribution, it is necessary to determine the opinion dynamics model, steady-state conditions, and the final opinion distribution. Finally, we turn to the problem with time-varying opinion space and determine through simulation how a time-varying opinion space with extremized opinions influences the steady-state opinion distribution.

1.3 Thesis Outline

Chapter 2 provides background on opinion dynamics and clustering. Clustering is the main method to analyze the result of the process of opinion dynamics. In detail, definition of opinion dynamics and clustering, opinion dynamics models, clustering techniques, and some applications on these topics are described in this chapter.

Chapter 3 details the opinion dynamics model and clustering methods used in this thesis. We introduce the DW model and check four clustering methods mentioned in chapter 2 to decide which one can be used with the DW model.

Chapter 4 describes the simulation design in MATLAB. First, we find an appropriate algorithm procedure to estimate the results of opinion dynamics. Then, scenarios for a two-dimensional opinion space and a time-varying opinion space are simulated with MATLAB.

Chapter 5 concludes this work including summary, contributions and future directions.

2 Background

This chapter introduces background information in terms of opinion dynamics and data clustering. The first section is a review of the literature on opinion dynamics, including its definition, models and applications. This is followed by some details on data clustering, which is necessary due to the complexity of the result of opinion dynamics. It also involves definition, applications, and related methods of data clustering.

2.1 Opinion Dynamics

Consider a group of agents with individuals capable of exchanging information or generally communicating about certain topics. “Each agent has an opinion about each topic which may change when it becomes aware of the opinion of others. Opinions are usually represented by real numbers in a continuous closed interval. This process of changing opinions is a process of continuous opinion dynamics” [1], and the Cartesian product of all opinion intervals is called opinion space. For example, consider a group of experts and citizens that collectively have to reach an agreement on the modification of tax rates. Experts are concerned about macroeconomic factors, and citizens care about their payment ability. Everybody has an opinion about the topic. To get a common agreement about the issue, they have to communicate and exchange opinions. We suppose that each agent is willing to revise his opinion by taking the opinions of other competent agents into consideration. We also suppose

there is a way of exchanging information between two agents and if the difference of opinions of two agents is not more than ε, they will change their opinions [4]. "ε is called the *bound of confidence*. This process of repeated discussing and revising of opinions is called *continuous opinion dynamics* under bounded confidence" [4].

2.1.1 Opinion Dynamics Models

Without an explicit formulation of a mathematical model, it is difficult to analyze the process of opinion formation. Many research studies have been carried out on this topic, including creating novel models, applications of a certain model, and stability and other features of a model. This section describes the classification of opinion dynamics models, which have three different categories: agent-based models and density-based models, discrete and continuous models, and homogeneous and heterogeneous models.

First, there are two types of models, agent-based model for a finite population of agents and density-based model for an infinite number of agents with a density function on the opinion space [5, 6].

Second, for the agent-based model of opinion dynamics, each agent can change opinion in time. Based on the situation of each agent's opinion, there also are two kinds of cases, the discrete case and the continuous case. Likewise, opinion dynamics

models may also separate into discrete opinion dynamics models and continuous opinion dynamics models. For discrete opinion dynamics models, they are models where the opinions are considered discrete [7]. In this group, there are the Ising model, voter model and Sznajd model [8-10]. On the second group, that of continuous opinions, the Deffuant and Weisbuch model (DW model) and the Hegselmann-Krause model (HK model) have received significant attention. Both of them apply the bound of confidence. It is a threshold that measures the difference between two interacting agents [7].

Finally, opinion dynamics models can also be divided into homogeneous and heterogeneous models. Every agent owns the same confidence level in homogeneous opinion dynamics model, whereas a heterogeneous bounded confidence model must be with social agents who may have diverse confidence levels because of complicated physiological factors [11]. In Ref [12], both DW and HK models are extended to heterogeneous bounds of confidence. The agents are grouped into close-minded and open-minded groups, based on their confidence levels. In Ref [13], a heterogeneous HK model was employed to give a theoretical convergence analysis under some assumptions on the existence of the equilibrium opinion vector and the time-invariant interaction topology. In Ref [14], a heterogeneous DW model was employed to give a convergence analysis under some assumptions.

2.1.2 Opinion Dynamics Applications

This section introduces the applications of opinion dynamics in a widely used area, including social systems, political decision making, robotics and other fields of science and engineering.

There is a vast literature concerning social systems or networks. Ref [15] applied opinion changing rate model to study whether and how a group of social agents, for example, with a various natural tendency to change opinion, finds agreement. Ref [16] gave a framework to analyze the controllability of opinion dynamics in social networks and described how the opinion can be controlled by a committed node. There also has been some research about the convergence of a model. For example, Ref [17] offered the study for convergence of majority model. Each agent approves the opinion which has the maximum social pressure based on this model.

There are many studies on political opinion dynamics. Ref [18] provided extensive discussions of the applications of opinion dynamics to political phenomena including "spatial organization, the formation of coherent structures (political parties), and the transition from unity to discord".

Recently, opinion dynamics is applied in robotic study. For example, Ref [3] studied the situation that an opinion dynamics model is capable of being the kernel of a collective decision-making mechanism for swarm robotics. The main result of this

research is that “when opinions represent action choices, the opinion associated with the action that is the fastest to execute spreads in the population. Moreover, the spread of the best choice happens even when only a minority is initially advocating for it.” In Ref [19], “an opinion formation model is used to capture important elements of the scenarios in which the proposed mechanism can be used in order to predict the system’s behavior. The model predicts that when the two actions have different average execution times, the swarm chooses with high probability the action with the shorter average execution time. The model’s predictions through a swarm robotics experiment in which robot teams must choose one of two paths of different length that connect two locations is validated.”

There are also some researches focusing on opinion dynamics under the influence of noise [20-22]. Ref [20] presented a study of Deffuant *et al* model which is affected by noise. An “order-disorder transition” is the main effect of noise. The opinion distribution tends to be uniform in the disordered state, whereas a number of opinion groups are generated in the ordered state.

For any determinate model, its features are also considered as research topics. The stability and convergence has been a popular study point for nearly each different model. Take the HK model as an example. Refs [23-26] provide convergence analysis based on the modified or original HK model. In Ref [24], “a stabilization theorem for processes of opinion dynamics is presented. The theorem is applicable to a wide class

of models of continuous opinion dynamics based on averaging. The analysis detects self-confidence as a driving force of stabilization." Besides, for a certain model, some of its parameters become the main study topics [27, 28]. Ref [28] is based on a model of opinion dynamics with decaying confidence. "This model is a multi-agent system where each agent receives the opinion of its neighbors and then updates its opinion by taking a weighted average of its own opinion and those of its neighbors that are within some confidence range. The confidence ranges are getting smaller at each time step."

2.2 Data Clustering

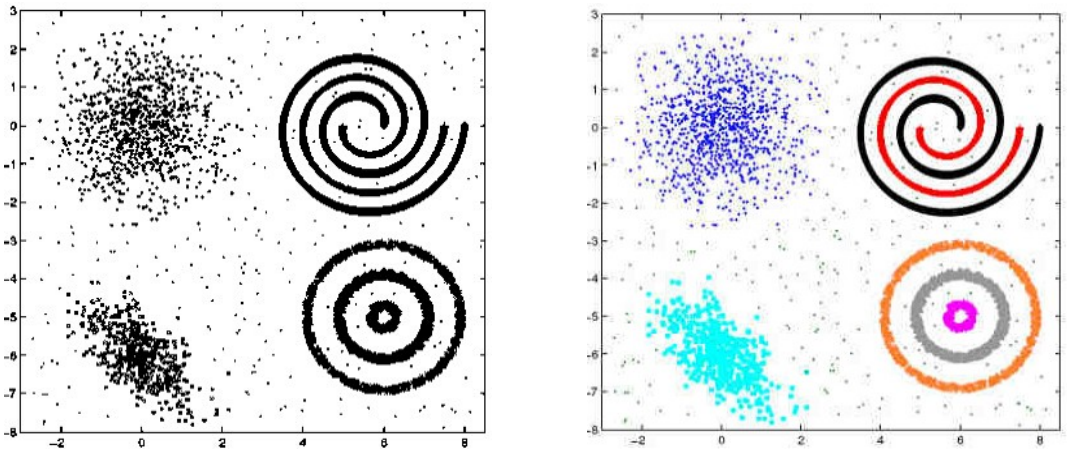
This section introduces data clustering because the results of the process of opinion dynamics do not always reach consensus. Taking the tax rate determination as an example, there might be more than two groups of experts or people who have varied opinions. Additionally, we can directly tell the clusters or groups in some cases, as most of them are one-dimensional or two-dimensional. But if the opinion space is three or more than three-dimensional, it's hard to identify the cluster number only by judgment of mankind. Therefore, we need tools or techniques to identify the results of opinion dynamics automatically.

As one of the most useful functions in the data mining process, clustering is applied for finding groups and distinguishing distributions and patterns in a set of data [29]. This section describes the definition of data clustering, its history, applications, and

several commonly used algorithms.

2.2.1 What is data clustering?

Data clustering (or cluster analysis) is aim to find the natural grouping(s) of a set of cases, data, or objects. Cambridge dictionary defines cluster analysis as “a way of studying or examining large amounts of data to find groups that are more like each other than they are like the data in other groups”. Here is an example of clustering in Figure 2.1.The aim is to find a method of discovering the natural groupings (Figure 2.1 (b)) in the unlabeled data (Figure 2.1 (a)) [30].



(a) Input data (b) Desired clustering

Fig. 2.1 Diversity of clusters. The seven clusters in (a) differ in shape, size and density. Although these clusters are apparent to a data analyst, none of the available clustering algorithms can detect all these clusters [30].

Based on the definition of clustering, it also can be defined as the following: given a set of data, find a number of groups according to a measure of similarity. However, as Fig. 2.1(a) shows, clusters can be different based on their shape, size, and density. It is

unclear for the definition of cluster. Besides, in some cases, noise in the data set can affect the detection of the clusters negatively. A set of compact and isolated points can be treated as an ideal cluster. In reality, we need automatic algorithms to find clusters because humans are not excellent cluster seekers for high-dimensional data. It is this challenge concerning the unknown number of clusters for a given data that has caused a number of clustering algorithms that have been published and that continue to appear [30].

2.2.2 Clustering methods and applications

Determining the number of clusters is one of the most difficult problems in data analysis. It has resulted in many clustering algorithms and techniques to address the problem. In this part, examples and applications of this work are introduced.

Examples of earlier and classical work are Calinski and Harabasz's index (1974), Krzanowski and Lai index (1985) and the silhouette method created by Rousseeuw in 1987 [31, 32]. In 1992, Celeux and Govaert built a parametric method, namely Bayesian Information Criterion, for a mixture of Gaussian distributions [33]. At the beginning of this century, several methods are proposed. Tibshirani reported gap statistic in 2001 [34]. In 2003, Sugar and James proposed the jump method. Prediction strength was published by Tibshirani and Walther in 2005. Recently, additional several methods have been proposed. Graph theory was used in Ref [35] for selecting the number of clusters. Another criterion for selecting the number of clusters is

meta-learning [36]. Although there are so many various approaches to estimate the number of clusters, FCM [37] and K-means [38] are also the commonly used methods. It is easy to get their algorithms in most popular mathematic tools such as MATLAB, so we are going to study clustering algorithms of those common methods which could be applied in later chapters with MATLB.

"Clustering problems arise in many different applications, such as data mining and knowledge discovery, data compression and vector quantization, and pattern recognition and pattern classification. The notion of what constitutes a good cluster depends on the application and there are many methods for finding clusters subject to various criteria" [39]. For the sake of understanding, we present some examples of application in clustering. First, the clustering algorithm can be used in identifying cancerous data sets. In Ref [40], the authors apply "k-means and Fuzzy c-means, two widely used clustering techniques, to cluster a lymph node tissue section which had been diagnosed with metastatic infiltration (cancer spread from its original location)." Second, clustering algorithms also play a key role in the area of search engines. According to Ref [41], image clustering was used to deal with image collections which are the most common and simple computer vision, image processing, and machine learning task. Based on cluster analysis, a system for analyzing students' result arranging their scores data according to the level of their performance is described in Ref [42]. In this paper, they implemented the k-mean clustering algorithm for analyzing students' result data. Third, clustering algorithms can also be

applied in wireless sensor network's based applications. One application is in Landmine detection. Ref [43] uses the clustering algorithm for data aggregation from sensor nodes reported from a high mine density region.

2.2.3 Clustering algorithms

There are many different algorithms related to cluster analysis. In order to choose a proper one for the opinion dynamics model that we are going to apply, in the following section, the most widely used clustering algorithms are chosen and described with details. The first technique is k-means clustering which was first proposed by James MacQueen in 1967 [44]. This technique depends on determining cluster centres by trying to minimize a cost function [45]. The second technique is Gaussian Mixture Models, which may be applied appropriate in the situation that clusters have different sizes and interconnection [46]. The third one is Fuzzy c-means clustering, which was first used by Bezdek in 1973 [47]. This technique is very similar to the earlier k-means clustering except that it assigns each data's point belonging to a cluster with a membership grade [45]. The last technique is subtractive clustering. This technique uses the positions of the data points to calculate a density function and chooses the position with the greatest density value as the centre of the clusters [45]. In the following, the main parts of descriptions of each clustering algorithm are cited from Ref. [45] to Ref. [49].

1. K-means clustering

Given a set of observations $\mathbf{x}_j$, $j = 1,...,n$, the objective of k-means clustering is to partition the n observations into c groups $\mathbf{G}_i$, $i = 1,...,c$ $(c \le n)$. $\mathbf{c}_i$ is defined as the centre of group $\mathbf{G}_i$. According to the Euclidean distance between an observation $\mathbf{x}_k$ in group j and the corresponding cluster $\mathbf{c}_i$, the cost function can be defined by the following function:

$$\mathbf{J} = \sum_{i=1}^{c} \mathbf{J}_i = \sum_{i=1}^{c} \left(\sum_{k, \mathbf{x}_k \in \mathbf{G}_i} \left\| \mathbf{x}_k - \mathbf{c}_i \right\|^2 \right) \tag{2.1}$$

where $\mathbf{J}_i = \sum_{k, \mathbf{x_k} \in \mathbf{G_i}} \left\| \mathbf{x}_k - \mathbf{c}_i \right\|^2$ is the cost function within group i.

A $c \times n$ binary membership matrix, $\mathbf{U}$, can define the partitioned groups. "Its element, u_{ij}, is 1 if the j^{th} data point, $\mathbf{x}_j$, belongs to group i, and 0 otherwise. Once the cluster centres, $\mathbf{c}_i$, are settled, the minimizing u_{ij} for Equation (2.1) can be derived as follows:

$$u_{ij} = \begin{cases} 1 & if \left\| \mathbf{x}_j - \mathbf{c}_i \right\|^2 \le \left\| \mathbf{x}_j - \mathbf{c}_k \right\|^2, for \quad each \quad k \ne i, \\ 0 & otherwise \end{cases} \tag{2.2}$$

which means that $\mathbf{x}_j$ belongs to group i if $\mathbf{c}_i$ is the closest centre among all centres.

On the other hand, if the membership matrix is fixed, i.e. if u_{ij} is fixed, then the optimal centre $\mathbf{c}_i$ that minimize Equation (2.1) is the mean of all vectors in group i:

$$\mathbf{c}_i = \frac{1}{\left| \mathbf{G}_i \right|} \sum_{k, \mathbf{x_k} \in \mathbf{G_i}} \mathbf{x}_k \tag{2.3}$$

where $|\mathbf{G}_i|$ is the size of $\mathbf{G}_i$, or $|\mathbf{G}_i| = \sum_{j=1}^{n} u_{ij}$.

The algorithm is presented with a data set $\mathbf{x}_j$, $j = 1,...,n$; it then determines the cluster centres, $\mathbf{c}_i$, and the membership matrix $\mathbf{U}$ iteratively using the following steps:

Step 1: Initialize the cluster centre $\mathbf{c}_i$, $i = 1,...,c$. This is typically done by randomly selecting c points from among all of the data points.

Step 2: Determine the membership matrix $\mathbf{U}$ by Equation (2.2).

Step 3: Compute the cost function according to Equation (2.1). Stop if either it is below a certain tolerance value or its improvement over the previous iteration is below a certain threshold.

Step 4: Update the cluster centres according to Equation (2.3). Go to step 2." [45]

The result of applying k-means clustering, as an example, is shown in Fig. 2.2. k-means clustering separates data into Voronoi-cells which is also known as Dirichlet regions.

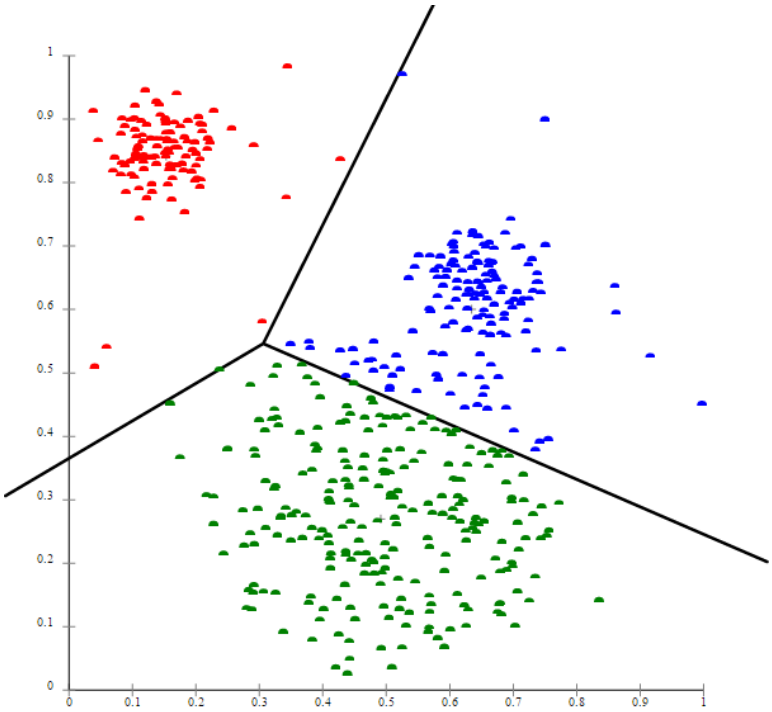


Fig. 2.2 K-means separates data into Voronoi-cells [48].

2. Gaussian Mixture Model

"A Gaussian Mixture Model is a parametric probability density function represented as a weighted sum of Gaussian component densities. Gaussian Mixture Model parameters are estimated from training data using the iterative Expectation-Maximization algorithm or Maximum A Posteriori estimation from a well-trained prior model" [49].

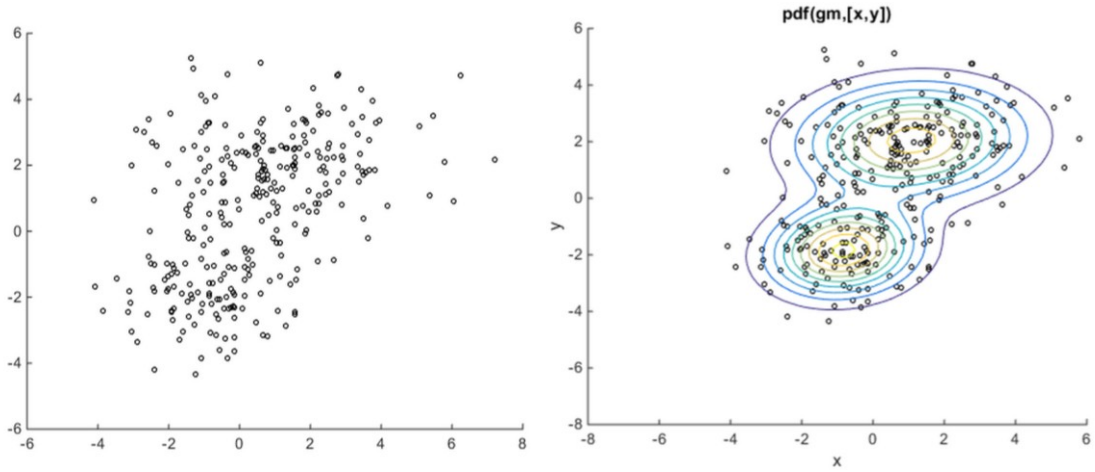
"Gaussian mixture models are often used for data clustering. Clusters are assigned by selecting the component that maximizes the posterior probability. Like k-means clustering, Gaussian mixture modeling uses an iterative algorithm that converges to a local optimum. Gaussian mixture modeling may be more appropriate than k-means clustering when clusters have different sizes and correlation within them. Clustering

using Gaussian mixture models is sometimes considered a soft clustering method. The posterior probabilities for each point indicate that each data point has some probability of belonging to each cluster.

A Gaussian mixture model is a sum of K component Gaussian densities as given by the equation,

$$p(x)=\sum_{k=1}^{K} p(k)p(x|k)=\sum_{k=1}^{K} \boldsymbol{\pi}_k N(\mathbf{x}|\boldsymbol{\mu}_k, \sum_k), \tag{2.4}$$

where $\boldsymbol{x}$ is a D-dimensional continuous-valued data vector, $\boldsymbol{\mu}_k$ is the mean vector, $\sum_k$ is the covariance matrix, $\boldsymbol{\pi}_k$, $k=1,...,K$, are the mixture weights which satisfy the constraint that $\sum_{k=1}^{K} \boldsymbol{\pi}_k = 1$"



(a) (b)

Fig. 2.3 Examples on Gaussian mixture model [46].

3. Fuzzy clustering

"Fuzzy C-Means (FCM) Algorithm generates fuzzy partitions and prototypes for any set of numerical data. These partitions are useful for suggesting substructure in

unexplored data ”[50].

“Fuzzy C-means clustering, relies on the basic idea of Hard C-means clustering, with the difference that in Fuzzy C-means each data point belongs to a cluster to a degree of membership grade, while in Hard C-means every data point either belongs to a certain cluster or not. So Fuzzy C-means employs fuzzy partitioning such that a given data point can belong to several groups with the degree of belongingness specified by membership grades between 0 and 1. However, Fuzzy C-means still uses a cost function that is to be minimized while trying to partition the data set ” [45].

“The membership matrix, $\mathbf{U}$, is allowed to have elements with values between 0 and 1. However, the summation of degrees of belongingness of a data point to all clusters is always equal to unity:

$$\sum_{i=1}^{c} u_{ij} = 1, \qquad (2.5)$$

$$\forall j = 1,...,n.$$

The cost function for Fuzzy c-means clustering is as follows:

$$\mathbf{J}(\mathbf{U}, c_1,...,c_c) = \sum_{i=1}^{c} \mathbf{J}_i = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m d_{ij}^2 \qquad (2.6)$$

where u_{ij} is between 0 and 1; $\mathbf{c}_i$ is the cluster centre of fuzzy group i; $d_{ij} = \|\mathbf{c}_i - \mathbf{x}_j\|$ is the Euclidean distance between the ith cluster centre and the jth data point; and $m \in [1, \infty)$ is a weighting exponent.

The necessary conditions for Equation (2.6) to reach its minimum are

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^{n} u_{ij}^m} \tag{2.7}$$

and

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}} \tag{2.8}$$

The algorithm works iteratively through the preceding two conditions until there is no more improvement noticed. In a batch mode operation, FCM determines the cluster centres, $\mathbf{c}_i$, and the membership matrix, $\mathbf{U}$, using the following steps:

Step 1: Initialize the membership matrix, $\mathbf{U}$, with random values between 0 and 1 such that the constraints in Equation (2.5) are satisfied.

Step 2: Calculate c fuzzy cluster centres $\mathbf{c}_i$, $i = 1,...,c$, using Equation (2.7).

Step 3: Compute the cost function according to Equation (2.6). Stop if either it is below a certain tolerance value or its improvement over the previous iteration is below a certain threshold.

Step 4: Compute a new $\mathbf{U}$ using Equation (2.8). Go to step 2." [45]

4. Subcluster

"The subtractive clustering method works as follows. Consider a collection of *n* data points, $\{x_1, x_2, ..., x_n\}$, in a *K*-dimensional space. Without loss of generality, the data points are assumed to have been normalized in each dimension so that they are

bonded by a unit hypercube. Each data point is considered as a potential cluster centre. The potential of data point $\mathbf{x}_i$ is defined as:

$$P_i = \sum_{j=1}^{n} \exp\left(-\frac{4\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{r_a^{\,2}} \right) \tag{2.9}$$

where, the symbol $\|\cdot\|$ denotes the Euclidean distance, and r_a is a positive constant. Thus, the measure of the potential for a data point is a function of its distances to all other data points. A data point with many neighboring data points will have a high potential value. The constant r_a is effectively the radius defining a neighborhood; data points outside this radius have little influence on the potential. After the potential of every data point has been computed, we select the data point with the highest potential as the first cluster centre. Let $\mathbf{x}_{c_1}$ be the location of the first cluster centre and P_{c_1} be its potential value.

$$P_i = P_i - P_{c_1} \exp\left(-\frac{4\left\|\mathbf{x}_i - \mathbf{x}_{c_1}\right\|^2}{r_b^{\,2}} \right) \tag{2.10}$$

Thus, we subtract an amount of potential from each data point as a function of its distance from the first cluster centre. The data points near the first cluster centre will have greatly reduced potential, and therefore will be unlikely to be selected as the next cluster centre. The constant r_b is effectively the radius defining the neighborhood which will have measurable reductions in potential. When the potential of all data points has been revised, we select the data point with the highest remaining potential as the second cluster centre. This process continues until a sufficient number of clusters are obtained. In addition to these criteria for ending the clustering process

are criteria for accepting and rejecting cluster centres that help avoid marginal cluster centres." [45]

5. Combining FCM with Subtractive Clustering

In terms of each algorithm discussed above, there exist problems while applying these techniques. Taking k-means and FCM clustering as examples, they are sensitive to the initial value which specifies their number of groups. If the number of clusters cannot be given properly before applying these techniques, the optimum solution will not be received. An algorithm of combining subtractive clustering with Fuzzy c-means clustering was created by W.Y. Liu in 2003 [51] to solve this problem. First, the optimal number of clusters and centres are estimated through subtractive clustering. The next step is to apply FCM algorithm to cluster the data set. There are many applications of combining these two methods. For example, to validate the performance and effectiveness of the proposed algorithm, Ref [52] applies the algorithm to Tennessee Eastman process and compares it with random initialization methods. The experimental results show that the proposed algorithm outperforms the other initialization methods. Ref [53] utilizes the algorithm to map the clustered observations from austempered ductile cast iron data. Ref [54] creates an improved Fuzzy c-means clustering algorithm which combines subtractive clustering and Fuzzy c-means clustering for color image segmentation.

2.3 Summary

Many papers have studied opinion dynamics in a large area with broad applications, including social science, engineering, computer science, and other research fields. In terms of features of each opinion dynamics model, research has been conducted widely concerning stability, convergence, and specific parameters. However, opinion dynamics with a time varying opinion space, which is the set of all possible opinions an agent may have, is a research topic that has barely been found in publications. Applications of the theoretical framework of opinion dynamics when the states are defined in shrinking opinion spaces include the modeling of a group of drones landing (from three-dimensions to two-dimensions), or a team of robots going across a narrow bridge (from two-dimensions to one-dimension). In the rest of this thesis, we are going to conduct research on this topic. The Deffuant-Weisbuch model, which is extensively used and allows modeling opinion dynamics in shrinking opinion spaces, is applied. The details of this model will be described in the next chapter.

In terms of data clustering, as we mentioned, the result of opinion dynamics does not always reach consensus. Sometimes, even for a small number of agents, the final number of groups can be larger than two. Besides, groups can differ in terms of their shape, size, and density. Without an accurate and clear standard for estimating clusters, it is mostly impossible to understand and analyze the result of groups related to opinion dynamics. Therefore, clustering is necessary and introduced in this chapter. With respect to clustering algorithms, we are going to test the four widely used

methods mentioned in section 2.23, which will be tested in MATLAB to discover which one is suitable with the Deffuant-Weisbuch model in the next chapter.

3 The Deffuant-Weisbuch Model and Assessment of Different Clustering Algorithms

This chapter introduces the Deffuant-Weisbuch model (DW model) for opinion dynamics and examines whether the four commonly used clustering algorithms, mentioned in chapter 2, work with the DW model for estimating the opinion formation. We first describe the fundamental definition and general application in the process of opinion dynamics. We then propose and analyze four clustering algorithms dealing with the opinion formation in the process of opinion dynamics related to the DW model. These algorithms are introduced in the form of functions in MATLAB.

3.1 The Deffuant-Weisbuch Model

The basic idea of the DW model is to determine the interactions among agents in which there is an exchange of opinion. If a subset of the opinions is close enough, the interaction results in the agents agreeing on the average of their initial opinions. Otherwise, opinions of the agents remain the same. By setting a multi-dimensional opinion space, it is therefore possible to make agents exchange opinion in topics in which they have eventually strong disagreement, provided that they agree enough on some (eventually) unrelated topic. By considering a thermo-mechanical analogy, the averaging between opinions resembles a dissipation process among colliding particles.

Due to the dissipative nature of the interactions, the dynamics of the system evolves into clusters that represent groups of individuals with homogeneous opinions within some given bound.

"Definition (Agent-based DW model) Suppose that there are m agents (or data points) and an appropriate opinion space, S. Given an initial profile, $x(0)$, and a bound of confidence $\varepsilon > 0$. In each time step, t, two random agents $i, j \in m$ perform the action

$$x^i(t+1) = \begin{cases} \frac{1}{2}\left(x^i(t) + x^j(t)\right) & if \left\| x^i(t) - x^j(t) \right\| \le \varepsilon \\ x^i(t) & otherwise \end{cases}, \qquad (3.1)$$

the same for $x^j(t+1)$ with i and j interchanged" [4] .

where $\|\bullet\|$ means Euclidean distance.

Here is the model with more details we are going to apply in the rest of this thesis. Given a number of agents, their opinions are two dimensional and spread throughout an opinion space which is the unit square with corners at (0, 0), (1, 0), (0, 1), (1, 1). This opinion space remains stable or shrinks at a certain rate, as simulated in the next chapter. The bound of confidence ε is between 0 and 1. Initially, all agents can choose their opinion randomly in the opinion space. Then, at each time step, randomly select two agents and measure their Euclidean distance. If their distance is not more than the bound of confidence ε, they will change their state to the mean of their previous values, otherwise they retain the same state. For example, choose agents i

and j, with opinions (0.3, 0.5) and (0.1, 0.8), respectively. Their distance equals 0.36. If ε is 0.4, they will both change their opinion to (0.2, 0.65). But if ε equals 0.3, their opinions will stay the same. A certain number of iteration steps later, all the agents might aggregate together. If the system is steady, any agent in one cluster cannot exchange opinion with other agents belonging to a different cluster. Clustering study is a crucial problem in opinion dynamics. It makes the processing and the result of opinion dynamics quantifiable. The next step is to assess the proper clustering algorithms which can be applied in the DW model.

3.2 Determine the Proper Clustering Algorithms

The four commonly used clustering methods and their basic mathematical foundations have been introduced in chapter 2. We then turn to the discussion of these techniques on the basis of a practical study. This study involves the implementation of each of the four techniques introduced previously, and testing each one of them on a set of data which comes from a result of opinion dynamics with the DW model.

Initially, given 1,600 agents randomly distributed in a unit square, with 0.3 as the confidence bound, after 45,000 iteration steps, we get Fig.3.1. It is clear that there are three clusters in this case. However, we need a proper technique to get the result automatically. The next step is to apply these four different techniques to partition these agents into clusters and find out the best one.

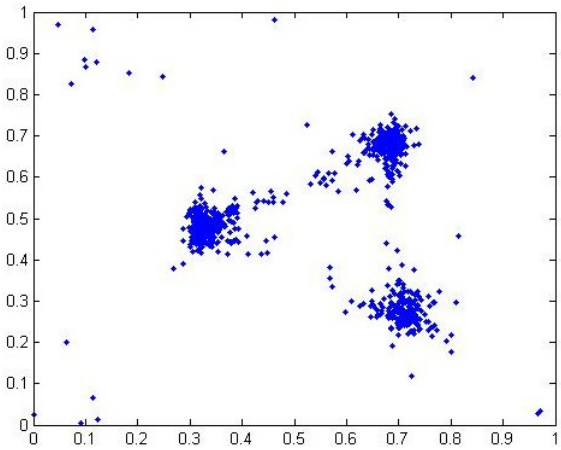


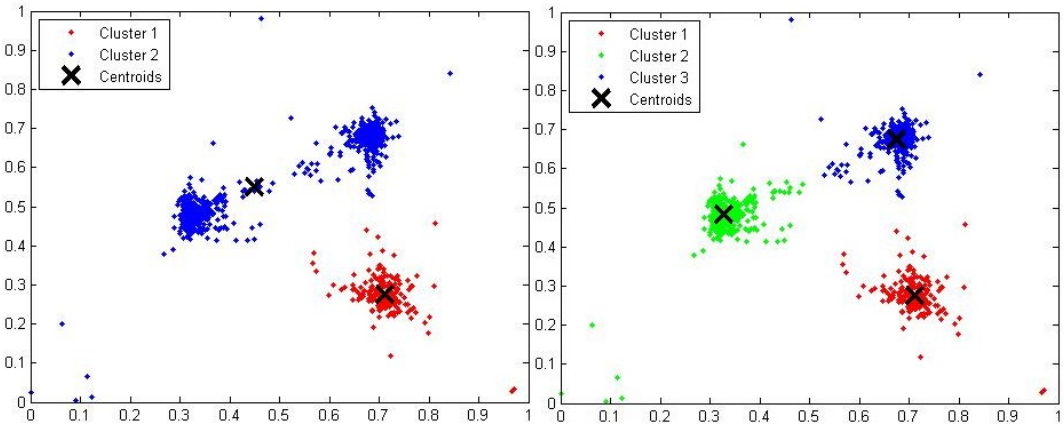
Fig.3.1 Initial condition of data points for testing clustering techniques

3.2.1 K-means

As mentioned in section 2.2.6, K-means aims to partition m agents into k clusters in which each agent belongs to the cluster with the nearest mean. In MATLAB, there is a function named *kmeans* which can fulfill the algorithm as follows. "*kmeans* treats each agent in the data set as an object having a location in space. It finds a partition in which objects within each cluster are as close to each other as possible, and as far from objects in other clusters as possible." We choose Euclidean distance as the metric. "Each cluster in the partition is defined by its member objects and by its centroid. Member objects are the objects belonging to a certain cluster. The centroid for each cluster is the point to which the sum of distances from all objects in that cluster is minimized "[55].

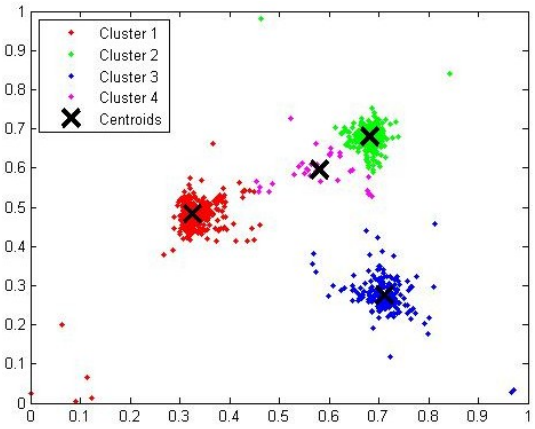
The syntax of kmeans in MATLAB is $[IDX, C, sumd, D] = kmeans(X, k)$. It returns distances from each point to every centroid in the n-by-k matrix D, the kth cluster centroid locations in the *k*-by-*p* matrix *C*, and an *n*-by-*1* vector *IDX* containing the

cluster indices of each point [55]. The following example explores possible clustering of initial condition in Fig. 3.1 by analyzing the results of partitioning the points into two, three, and four clusters.



(a)

(b)



(c)

Fig.3.2 Partitioning data by using kmeans function. a) when k equals 2; b) when k equals 3; c) when k equals 4

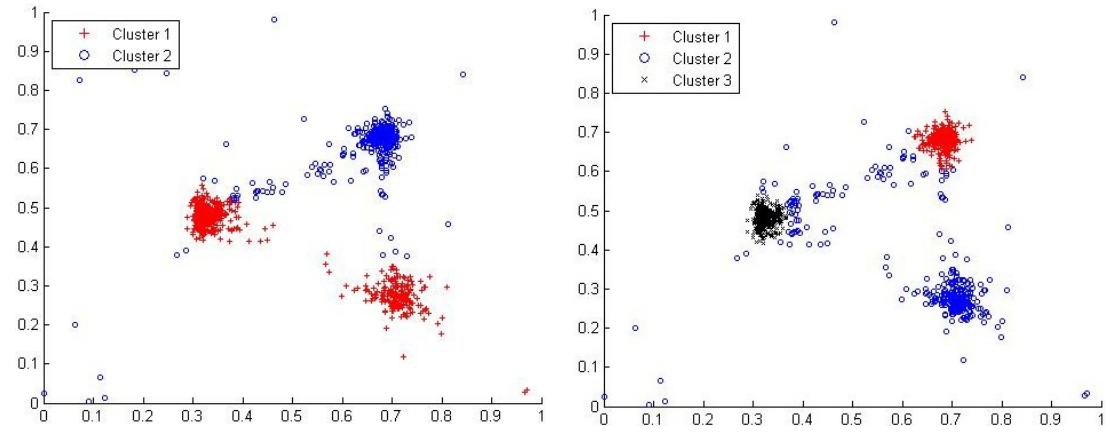
It shows that the best way to partition data is when k equals 3. However, this technique cannot provide the right number of clusters automatically. We have to identify the right number of clusters before applying the method or try several values of k and figure out which one is the best based on their results as we did. This

problem will be discussed in chapter 4.

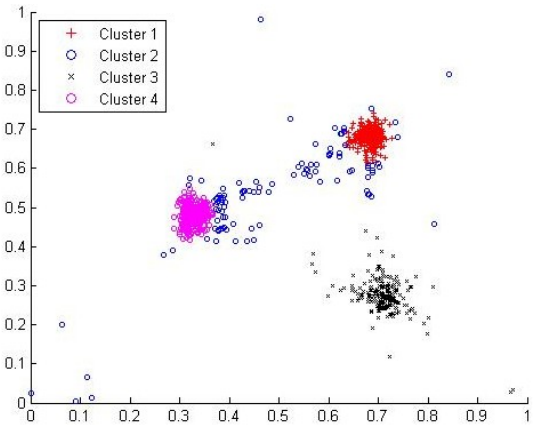
3.2.2 Gaussian Mixture Models

"Gaussian Mixture Models are formed by combining multivariate normal density components. Similar to k-means clustering, Gaussian mixture modeling uses an iterative algorithm that converges to a local optimum. Gaussian mixture modeling may be more appropriate than k-means clustering when clusters have different sizes and correlation within them" [48]. In MATLAB, the function *fitgmdist* is used to fit a Gaussian mixture distribution to data. For example, $GMModel = fitgmdist(X, k)$ returns a Gaussian mixture distribution model (GMModel) with k components fitted to data *X*.

We adopt the same data used in the example of k-means clustering, and then apply the function *fitgmdist* to return a Gaussian mixture distribution model with two, three and four components. The results are:



(a) (b)



(c)

Fig.3.3 Partitioning data by using GMM method. a) when k equals 2; b) when k equals 3; c) when k equals 4

As in the method of *K-means,* we have to provide the value of k before applying this method. Additionally, Gaussian Mixture Models technique is sensitive to noise and cannot provide a suitable partitioning for the distribution of data points generated with the DW model.

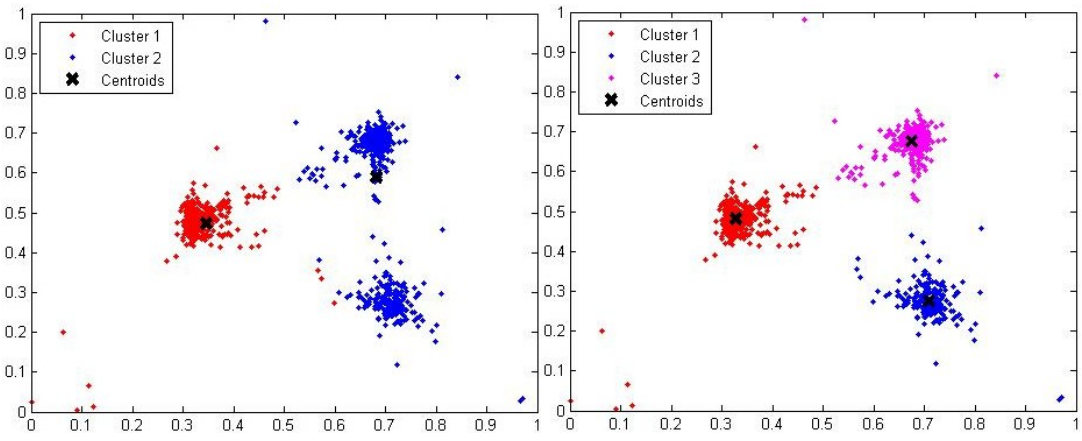
3.2.3 Fuzzy C-means Clustering

Fuzzy C-means Clustering is an improvement over earlier Hard C-means clustering. In this technique each data point belongs to a cluster to a degree specified by a

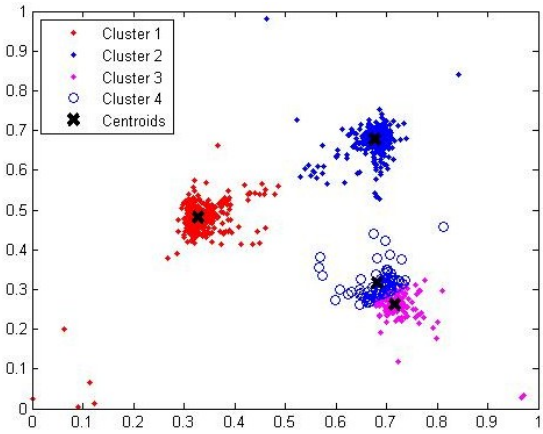
membership grade. In MATLAB, the command line function *fcm* outputs a list of cluster centres and several membership grades for each data point. We can use the information returned by *fcm* to build a fuzzy inference system by creating membership functions to represent the fuzzy qualities of each cluster.

The syntax of this function is $[center, U, objFcn] = fcm(fcmdata, k)$. Here, "the variable *centre* contains the coordinates of the k cluster centres, U contains the membership grades for each of the data points, and *objFcn* contains a history of the objective function across the iterations" [56] .

Similar to the example related to *kmeans*, we explore possible clustering of initial condition in Fig. 3.1 by analyzing the results of partitioning the points into two, three, and four clusters.



(a) (b)



(c)

Fig.3.4 Partitioning data by using FCM. a) when k equals 2; b) when k equals 3; c) when k equals 4

We get a proper result which is similar to that of k-means when the value of k equals three. Also, it has the same problem as the k-means as the number of clusters has to be assigned a priori. Compared to k-means clustering, Fuzzy c-means clustering can provide the probability of each agent belonging to related clusters.

3.2.4 Subtractive Clustering

If we do not know a priori how many clusters there should be for a given set of data, subtractive clustering is a fast, one-pass algorithm for estimating the number of clusters and their centres in a set of data. The syntax of the MATLAB function that implements it is $[C,S]=subclust(X,radii,xBounds,options)$. It returns the matrix, C, that contains clusters' centres. The vector S contains the sigma values that specify the range of influence of a cluster centre in each of the data dimensions. For its input, X is the matrix which contains the data to be clustered. Radii are vectors

of entries between zero and one that specify a cluster centre's range of influence. Small radii values generally result in finding a few large clusters [57]. Here are the results of applying different radii.

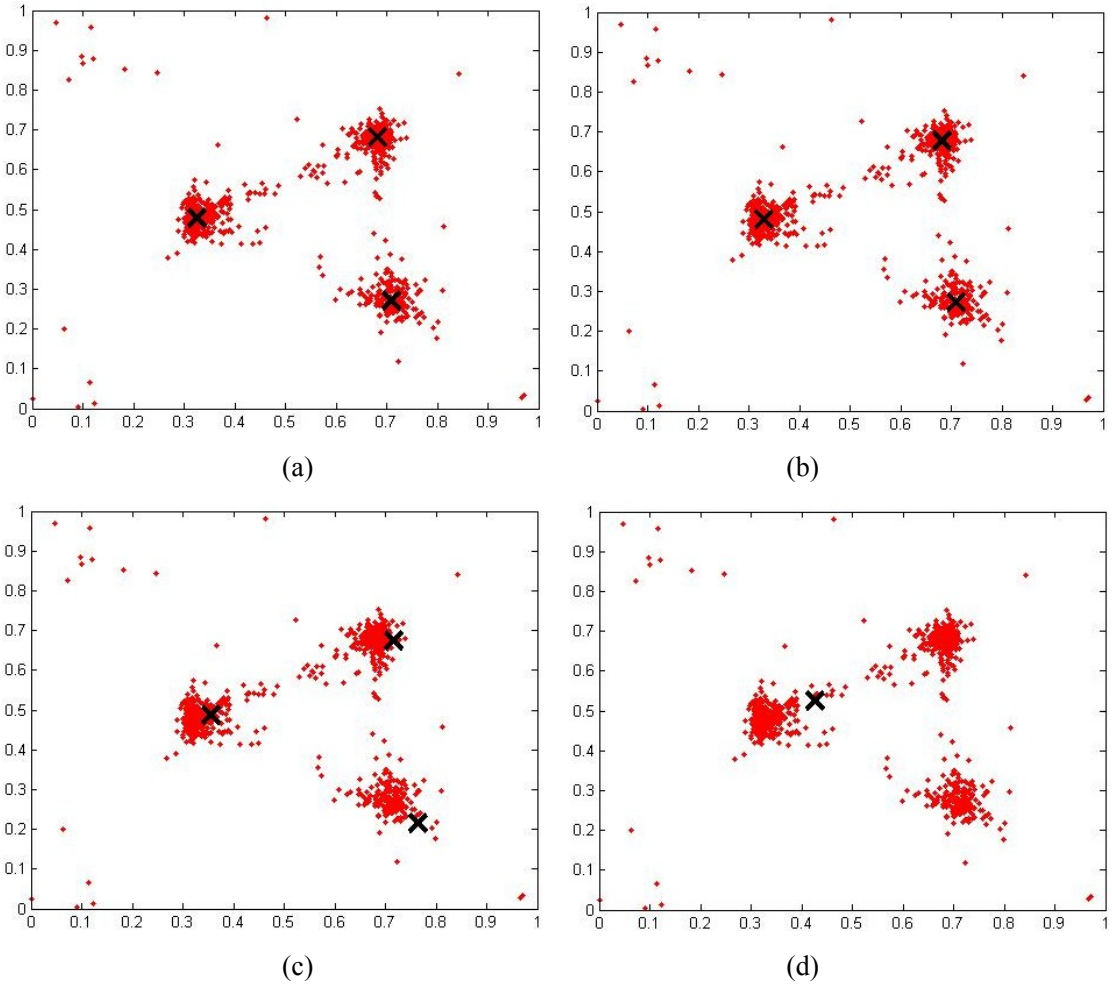


Fig.3.5 Partitioning data by using Subtractive Clustering. a) when radii equals 0.1; b) when radii equals 0.3; c) when radii equals 0.5; d) when radii equals 0.7

With the *subcluster* function, we can easily get the number of clusters and the location of centres. However, this method does not return agents memberships to clusters. Unlike other clustering techniques we mentioned, *subcluster* function cannot provide the details of each cluster, including how many agents (or data points) there are in a

cluster and where they are. In the next chapter we will show how we combine two algorithms to estimate number of clusters, their centres, and points' membership grades. This will ultimately allow deciding if the evolution of the system has reached steady state.

3.3 Summary

This chapter describes the basic definition and a practical example of the DW model, which also gives details to the application of this model concerning two-dimensional or shrinking opinion spaces. Four clustering algorithms are introduced in the form of functions in MATLAB. Each of them is conducted and three of them can be applied in analyzing the result of opinion dynamics based on the DW model. However, these available algorithms have their own problem when estimating the clustering or analyzing the convergence of the system (will be discussed in chapter 4). More specifically, *k-means* and *Fuzzy c-means* clustering performed well in partitioning the data into clusters, but the proper number of clusters has to be given before applying. Gaussian Mixture Models clustering is not suitable for the case based on the DW model because it is affected by noise which is some data points far away from the main aggregation. The distance of some noises from the main aggregation is larger than the bound of confidence. Subtractive clustering is suitable for determining the number of clusters when this parameter is unknown, but it does not return membership grades of data points with respect to clusters. In the next chapter, we will

combine different clustering algorithms to be able to determine number of clusters and data points membership grades.

4 Study on Clustering and Opinion Dynamics Related to the DW Model

This chapter is dedicated to the study of simple opinion dynamics scenarios based on the DW model discussed in chapter 3. Different scenarios are simulated to study the influence of characteristic parameters on the formation of opinions, which is quantified by the formation of clusters in the opinion space. Additionally, we simulate the scenario of a two-dimensional opinion space in which one side shrinks, and evaluate how the rate of shrinking influences the steady-state opinion space. This is a simplified model to gain some insight on the effect of the extremization of opinions in multi-dimensional opinion spaces.

As a preamble for simulations of opinion dynamics, the next section is dedicated to the discussion of the clustering technique that is used to estimate the number of clusters in the opinion space. This is a crucial step, as the steady-state number of clusters characterizes the opinion dynamics when individuals in different clusters cannot communicate anymore, as the distance (with respect to a suitable metric) between pairs of clusters is larger than a minimum distance considered acceptable to change the opinions. Details can be found in the discussion of the DW model in section 3.1.

4.1 Estimate the number of clusters

In this section we propose a method to estimate the number of clusters that combines Fuzzy c-means clustering and subtractive clustering. The details of these two methods have been discussed in chapter 3.

In fact, we can choose either Fuzzy c-means or k-means clustering to combine with subtractive clustering. Both of them are capable of partitioning data points into different clusters. The difference between the two methods is that k-means is deterministic as it assigns each data point to a cluster, whereas Fuzzy c-means assigns to each data point a probability distribution defined on the clusters, so that for a data point it is possible to know the probability that it belongs to different clusters. Here we adopt Fuzzy c-means and assign points to clusters by considering the highest probability.

Fuzzy c-means clustering is a method for partitioning data into clusters that has the important feature of assigning the probability of each data point belonging to every cluster (membership grades). However, this technique suffers from the necessity of having to declare the number of clusters. On the other hand, subtractive clustering is suitable to estimate the numbers of clusters in a data set, but it is affected by convergence issues when one tries to use it to determine the relation between data points and clusters, as for example in terms of probability.

By combining the two techniques we can overcome the limitations of the two. First, we use subtractive clustering to find the number of clusters. Then, given the estimated number of clusters, Fuzzy c-means clustering is applied to assign data points to clusters and to find out each cluster's location. With the information offered by Fuzzy c-means clustering, we can determine if the system is at steady state. If the system has not reached steady state, the simulation continues until the next check. The process is schematized in Fig.4.1 in which the iteration is explicitly represented in terms of blocks associated with fundamental actions. Details of the steps involved in the determination of the eventual steady state (convergence analysis) are discussed in the next section.

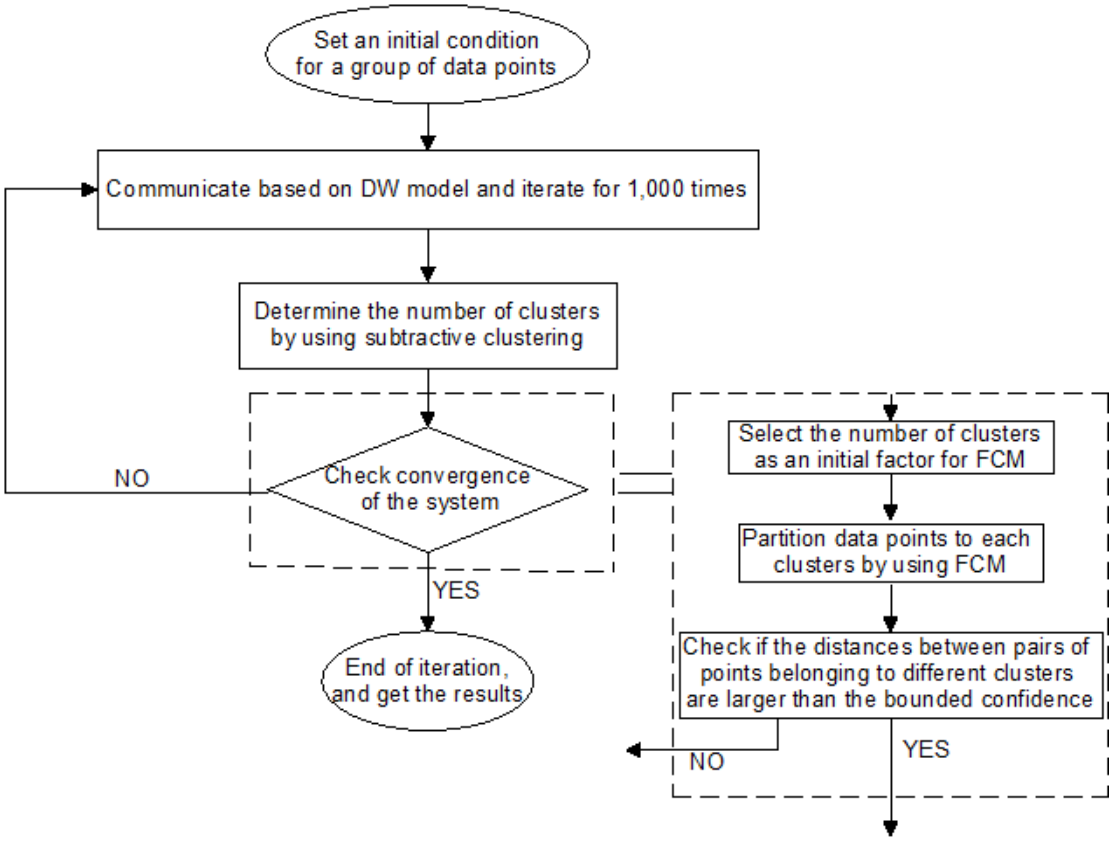
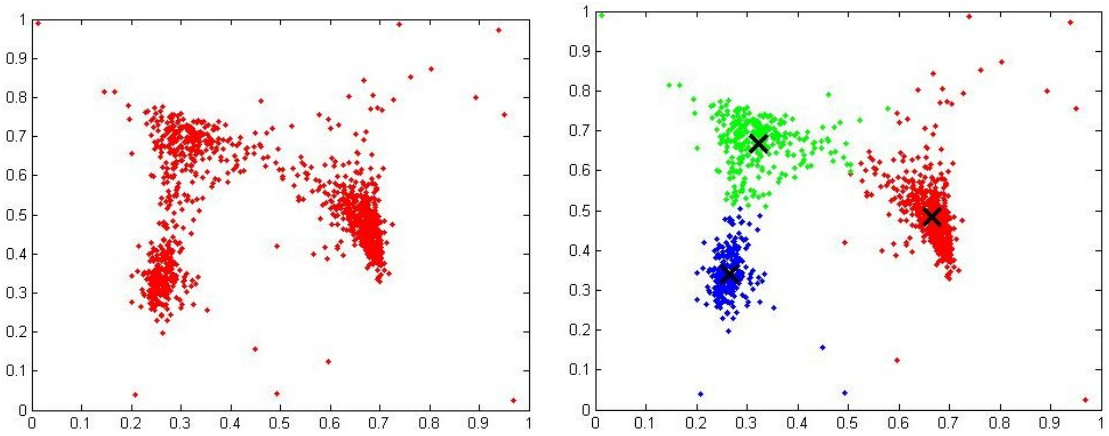


Fig.4.1 Algorithm procedure for ascertaining the number of clusters

4.1.1 Steady State and Verification of the Exit Condition

The opinion dynamics space is stationary when points in each cluster can communicate only with points within the same cluster, where the possibility of communication is quantified by a distance parameter that sets which is the minimum distance between opinions so that communication/opinion exchange between two agents results in changing their initial opinions. In this case we also say that convergence of the system is achieved. To check the convergence, we check the distances of every pairs of points belonging to different clusters. If there exists one pair of points such that their distance is less than the bounded confidence, the system has not reached convergence; otherwise the steady state of the system is achieved.

The following example is constructed for the purpose of checking steady state by running a simulation with the DW model. Given 1,600 two dimensional data points randomly in a box of one by one, 0.3 as the confidence bound and running 40,000 iterations, we obtain the configuration in Fig. 4.2 (a). The method of subtractive clustering tells that there are three clusters in Fig. 4.2 (a). In order to analyze data points, we have to determine their relation to clusters, so Fuzzy c-means clustering is run. In this way, we assign a membership grade to every data point. In this case, each point will have three membership grades assigned: the largest one determine the belonging to a certain cluster. Fig. 4.2 (b) shows the belongings of each data point.



(a) (b)

Fig.4.2 Result after 40,000 iterations running. a) Distribution of data points. b) shows the belongings of each data point.

1. Steady state

The criterion adopted is that if the distance between any pair of points belonging to different clusters is larger than the bounded confidence, then the system has reached steady state as these points cannot exchange opinions anymore (within the DW model). For the specific example, the system in Fig. 4.2 was not steady state. The steady state system for the confidence bound 0.3 is represented in Fig. 4.3, and it has only one cluster.

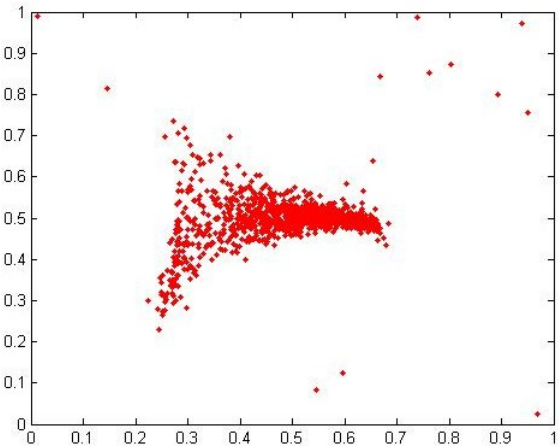


Fig.4.3 Result after 9,000 iteration steps.

Comparing Fig.4.2 with Fig.4.3, it is clear that there are points that do not change

their state even after a very large number of iterations since their distance from the main aggregations is larger than the confidence bound. Such points are considered outliers. The number of outliers may or may not be relevant with respect to the size of the data population. If the number of outliers is sufficiently small, they are simply discarded in the clustering process; on the other hand, if the number is sufficiently large, then the clustering process must account for them.

Outliers and how to account for them will be discussed in the next section.

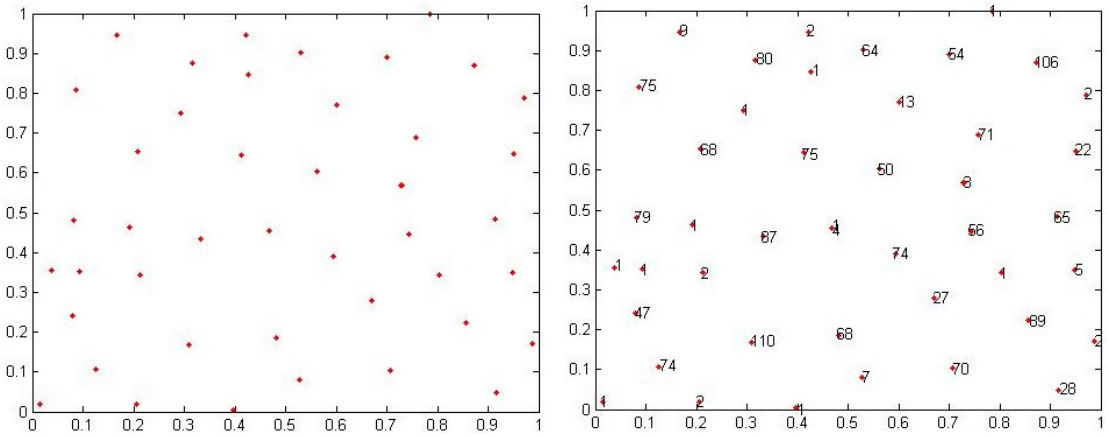
4.1.2 Outliers and Confirmation of the Number of Clusters

Due to the absence of an accurate and clear definition of clusters and outliers, there exists the difficult problem of distinguishing clusters and outliers when the number of outliers is sufficiently large. In this section, we are going to discuss this problem and try to solve it by using silhouette statistic and subtractive clustering.

1. **Outliers**

As mentioned, outliers are data points whose distances from the main aggregations are larger than the confidence bound. In the case of Figs. 4.2 and 4.3, the number of outliers is not large enough to be treated as a cluster. So they are discarded in the process of clustering. However, when the number of outliers is considerable large, it is necessary to consider the influence of outliers on clustering. In order to make this problem clearer and more understandable, we take an example as follows.

Given 1,600 two-dimensional data points randomly distributed in a one by one box, 0.1 as the confidence bound and taking communication until the system reaches steady state, we can get a result shown in Fig. 4.4 (a).



(a) (b)

Fig.4.4 An overlapped case when 0.1 is the confidence bound; (a) shows the result in a steady state (b) shows the number of data points

In this case the simulation has run long enough so that clusterized data points are overlapped by assuming much closed states. In this way, it is difficult to distinguish outliers and clusters. To solve this problem, we determine the size of each cluster and show the result in Fig. 4.4 (b). Even if we know the degree of overlapping (the number of data points in each overlapped place), this problem has not even been solved because the degree of overlapping for a determined cluster is uncertain. Thus, we need a better way to determine the correct number of clusters. In the following, silhouette statistic and subtractive clustering will be discussed to get a solution.

2. Silhouette statistic

In this section we cite the basic conception of silhouette statistic from Ref [58]. “We

note that the definition of a cluster depends on the application and it is not always clear what should be the optimal number of clusters for a given problem. The usual approach to solve this problem is to define a parametric model for the shape of clusters or to use a two-step procedure where a clustering algorithm is applied and then determines the number of clusters. We follow the latter approach and use the silhouette statistic proposed by Rousseeuw (1987) as the goodness of classification measure."

Let $X = \{x_1,...,x_n\}$ be a data set with n elements and let $d(x_i, x_j)$ denote the distance between x_i and x_j. The Euclidean distance is the most common choice. Suppose that we must classify each element of the data, X, in one of the following k clusters, $C^1, C^2,..., C^k$.

Define

$$d(x_i, B) = \frac{1}{\#B}\sum_{x \in B} d(x_i, x), \tag{4.1}$$

as the average dissimilarity of x_i to all elements of cluster B, where #B is the number of elements of B. Denote by A the cluster to which x_i has been assigned by the clustering algorithm and by C any other cluster different of A. Define

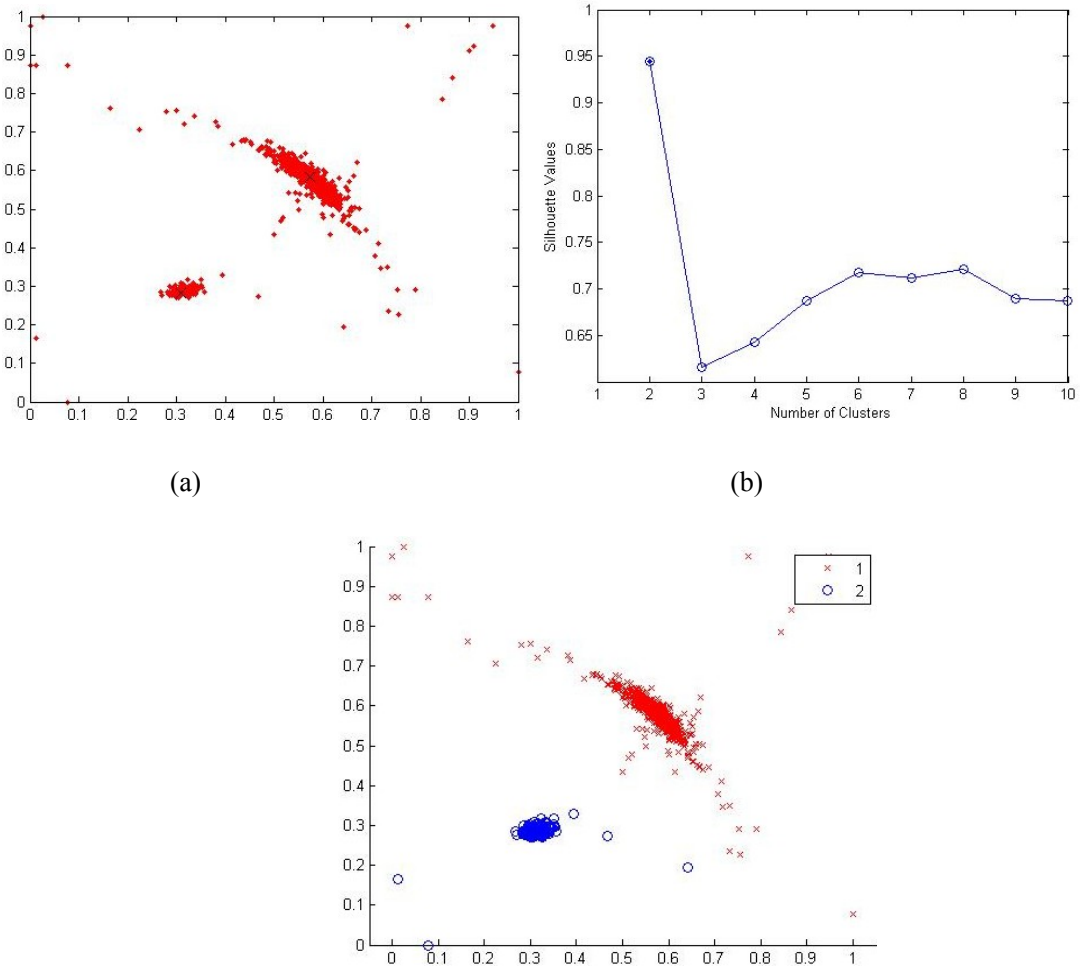
$a_i = d(x_i, A)$ and $b_i = \min_{C \neq A} d(x_i, C)$

The quantities a_i and b_i are the "within" dissimilarity and the smallest "between" dissimilarity, respectively. Then a proposal to measure how well object x_i has been clustered is given by Rousseeuw (1987)

$$s_i = \begin{cases} \dfrac{b_i - a_i}{\max\{b_i, a_i\}}, & if\ \#A > 1 \\ 0, & if\ \#A = 1 \end{cases} \tag{4.2}$$

It is easy to see that $-1 \le s_i \le 1$. If s_i is closed to 1, we know that datum is appropriately clustered. If s_i is close to negative one, we see that i would be more appropriate if it was clustered in its neighboring cluster. If s_i is near zero, it means that datum is on the border of two natural clusters. [58]

The data used in the following example was taken from a simulation by giving an initial condition of 1,600 two-dimensional data points randomly distributed, and 0.3 as the confidence bound. By letting the system evolve based on the DW model, and until the system reaches convergence, we get the picture Fig.4.5 (a). Then, we evaluate the optimal number of clusters using the silhouette clustering evaluation criterion with MATLAB and get a result shown in Fig. 4.5 (b). The plot shows that the highest silhouette value occurs for two clusters, suggesting that the optimal number of clusters is two. In Fig. 4.5 (c), we can tell that the plot shows two distinct clusters within the data: cluster 1 is in the upper-right corner, and cluster 2 is in the lower-left corner. We also test silhouette method with different confidence bound values and find that this method can evaluate the optimal number of clusters well in most cases. But it is not suitable in some cases when the data points are all overlapping.

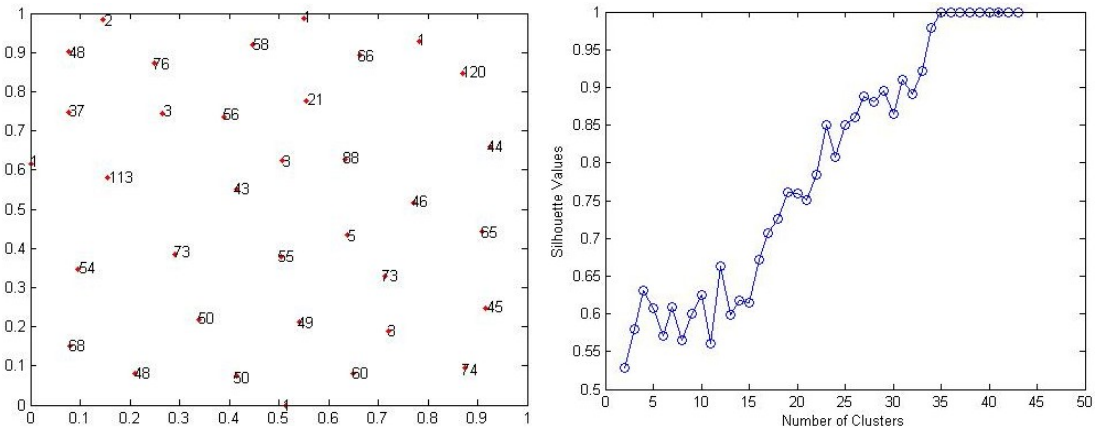


(a) (b)

(c)

Fig.4.5 An example of silhouette method. (a) is the original data. (b) shows the silhouette criterion values for each number of clusters tested. (c) creates a grouped scatter plot to visually examine the suggested clusters.

We now test the silhouette method in an overlapping data case. Given the same initial conditions, 0.1 as the confidence bound, the result is shown in Fig.4.6 (a).



(a) (b)

Fig.4.6 An example of silhouette method in an overlapping data case. (a) is an original data (b) shows the silhouette criterion values for each number of clusters tested

It shows that the silhouette method is improper for the overlapping data case. If we determine points whose number is less than 10 as outliers, there should be 26 clusters in Fig.4.6 (a). However, Fig.4.6 (b) tells that the silhouette method provides 41 clusters for the optimal number of clusters. It is even larger than the number of overlapping points we can see in Fig.4.6 (a), so it is not suitable for this case.

In conclusion, the silhouette method can be applied in the case where data points are not overlapped.

3. Subtractive Clustering

As mentioned in Section 3.2.4, subtractive clustering is a fast, one-pass algorithm for estimating the number of clusters, especially when the potential density value (in Equation (2.9)) is the main standard to be taken into account. Moreover, it is clear that overlapping data is a problem in terms of density because if the degree of overlapping is large, the probability for an aggregation that can be treated as a cluster is large.

Thus, theoretically, subtractive clustering should perform well in this case. We apply subtractive clustering with radii equals 0.1, which specifies very small cluster centre's range of influence. Then we get the result in Fig. 4.7, which shows that there are 26 clusters. Compared to Fig. 4.6 (a), it is more reasonable because all the assigned clusters have a large number of points. Therefore, in the overlapped data case, subtractive clustering is a good method to apply.

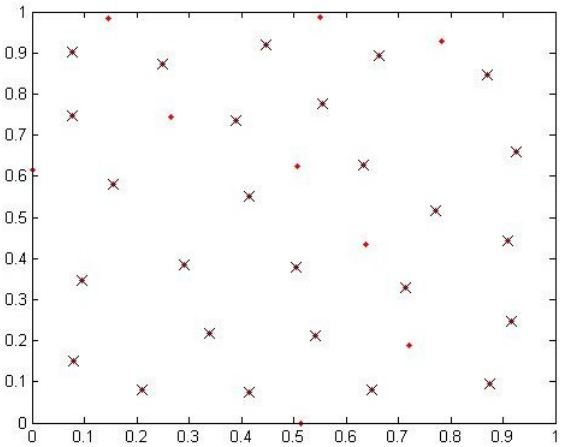


Fig.4.7 An example of Subtractive Clustering in an overlapped data case

4.2 Simulation with DW Model

This section is dedicated to the simulation of opinion dynamics with the DW model in different scenarios. The influence of characteristic parameters on the formation of opinions is studied, such as initial conditions, the bounded confidence and the number of data points. This is a general view on the effect of parameters on the formation of clusters in the opinion space. Additionally, a time varying opinion space which shrinks from two-dimensional space to one dimensional space with different rates is investigated.

4.2.1 Initial Condition: Randomly Distributed Data

In this section, two-dimensional cases with an initial condition given by randomly distributed data points are simulated by MATLAB. As shown in Fig.4.8, initially there are 1,600 data points randomly distributed in a box which is one by one unit length. They communicate with each other randomly and change their position based on the DW model in Equation (3.1). When the system reaches convergence, which means there will be no communication between any points belonging to different clusters, the simulation stops. Then, we get the final number of clusters and the formation of opinions.

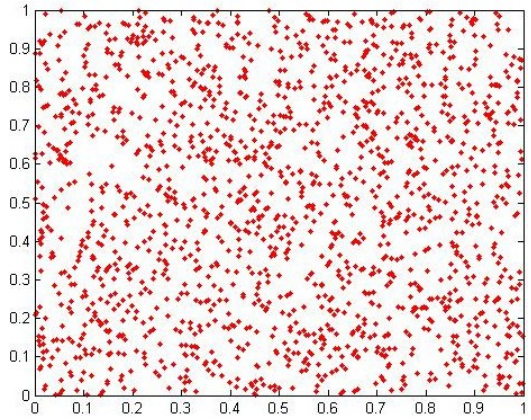
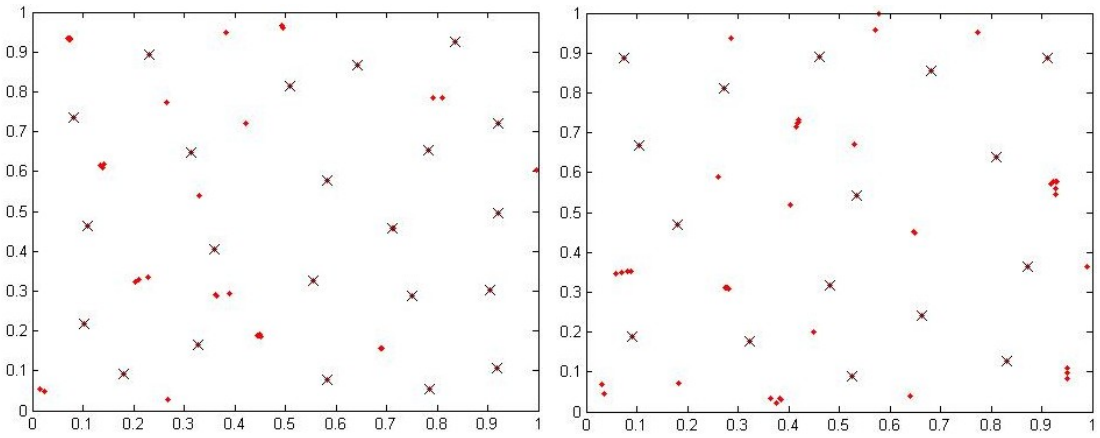


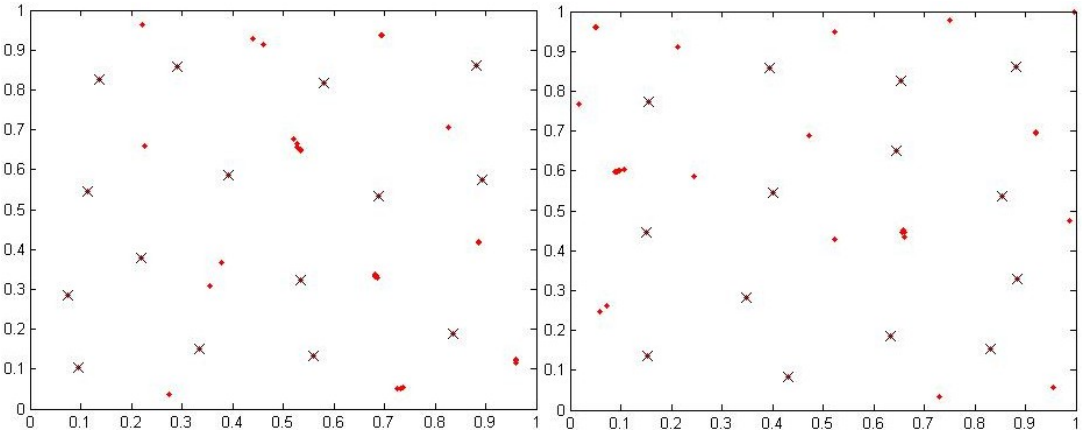
Fig.4.8 Initial condition with randomly distributed data

Even if the initial conditions are the same, the results can be different for every running, since communication events between data points are random. Therefore, to form a statistics, simulation is performed ten times for each bounded confidence. The average number of clusters for each ε is calculated and reported in the results. In the following figures, red points are data points and black crosses are the centres of

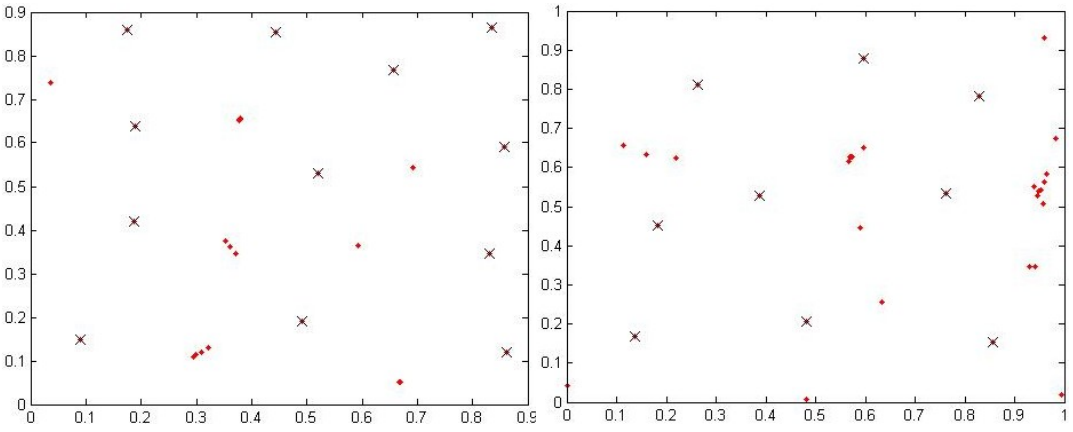
clusters.



(a) (b)



(c) (d)



(e) (f)

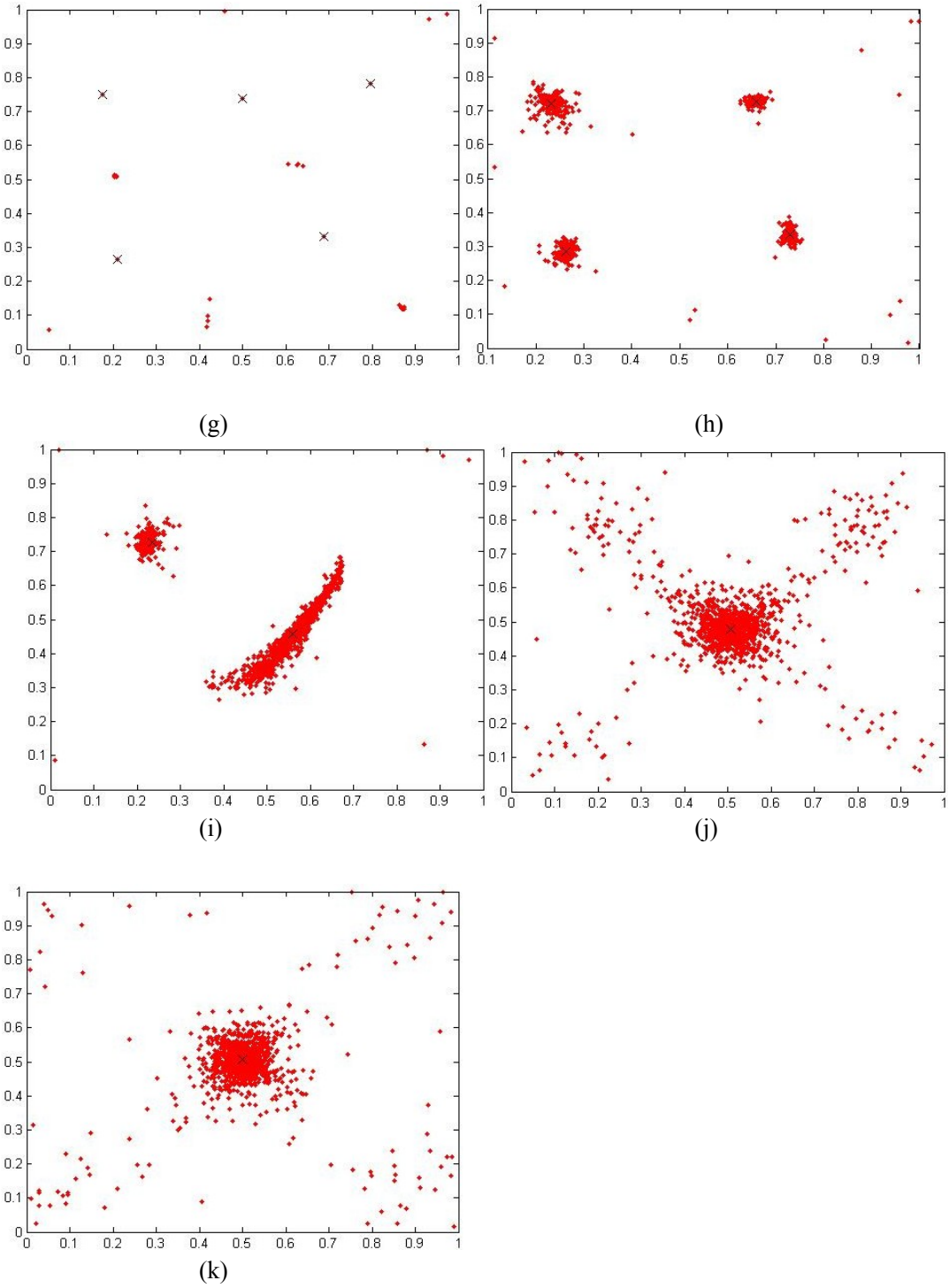


Fig.4.9 Simulation with the DW model in two-dimensional cases. (a) ε equals 0.1. (b) ε equals 0.11. (c) ε equals 0.12. (d) ε equals 0.13. (e) ε equals 0.14. (f) ε equals 0.15. (g) ε equals 0.2. (h) ε equals 0.25. (i) ε equals 0.3. (j) ε equals 0.35.(k) ε equals 0.4.

Table 4.1 Average number of clusters under the initially randomly-distributed data condition

ε	0.10	0.11	0.12	0.13	0.14	0.15	0.20	0.25	0.30	0.35	0.40
Number of clusters	23	18	15	14	11	9	5	4	2	1	1

Table 4.1 and Fig. 4.10 show that as the bound of confidence increases, the final number of clusters declines in the initial condition that data points are randomly distributed. We also get a result that when the bound of confidence is larger than 0.35, the final number of clusters reduces to one.

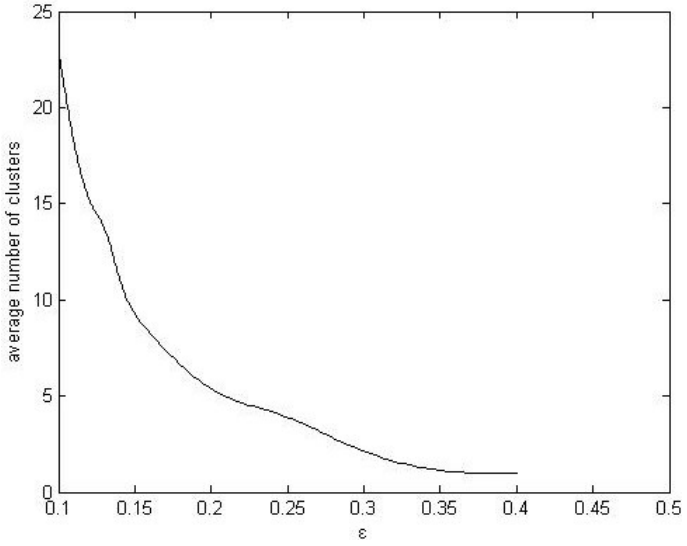


Fig.4.10 Average number of clusters versus ε

4.2.2 Initial Condition: Regularly Distributed Data

In this section, two-dimensional cases with the initial condition of regularly distributed data points are simulated by MATLAB. The initial condition is shown in Fig. 4.11, The system evolves according to the DW model in Equation (3.1). Until the system reaches convergence, which means there will be no communication between

numbers of different clusters, and the simulation stops. The last step is to estimate the number of clusters.

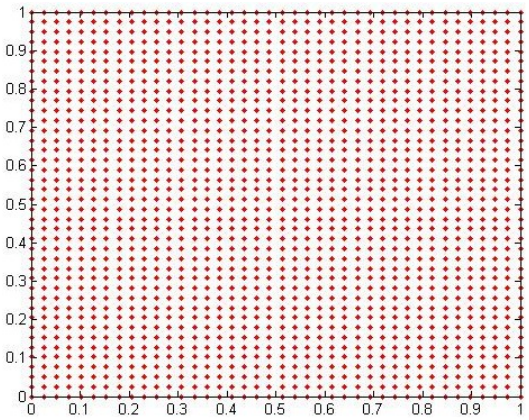
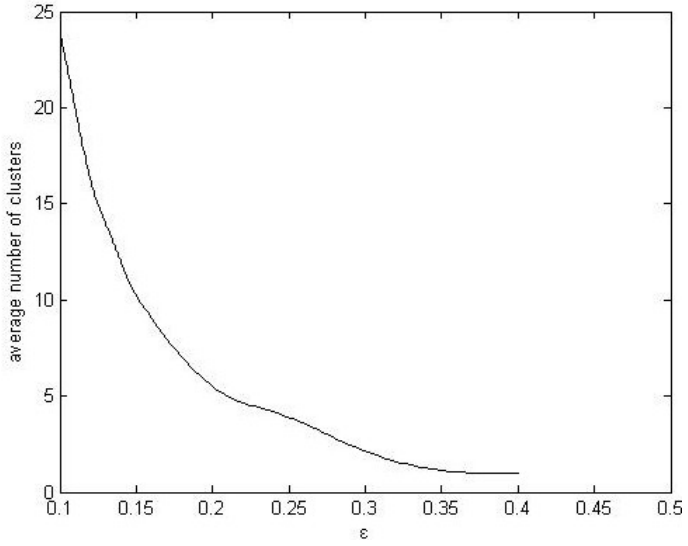


Fig.4.11 Initially regularly distributed data

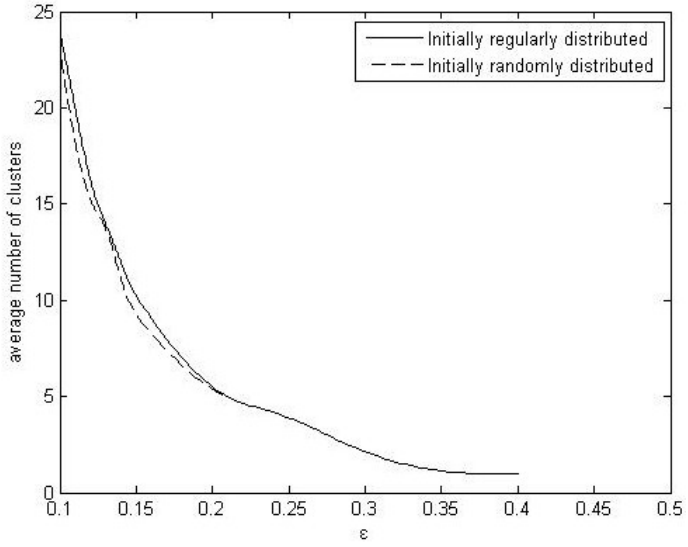
We perform the simulation ten times for each bound of confidence and finally get their average number of clusters.

Table 4.2 Average number of clusters under the initially regularly-distributed data condition (ten runs)

ε	0.10	0.11	0.12	0.13	0.14	0.15	0.20	0.25	0.30	0.35	0.40
Average	24	20	16	14	12	10	5	4	2	1	1



(a)



(b)

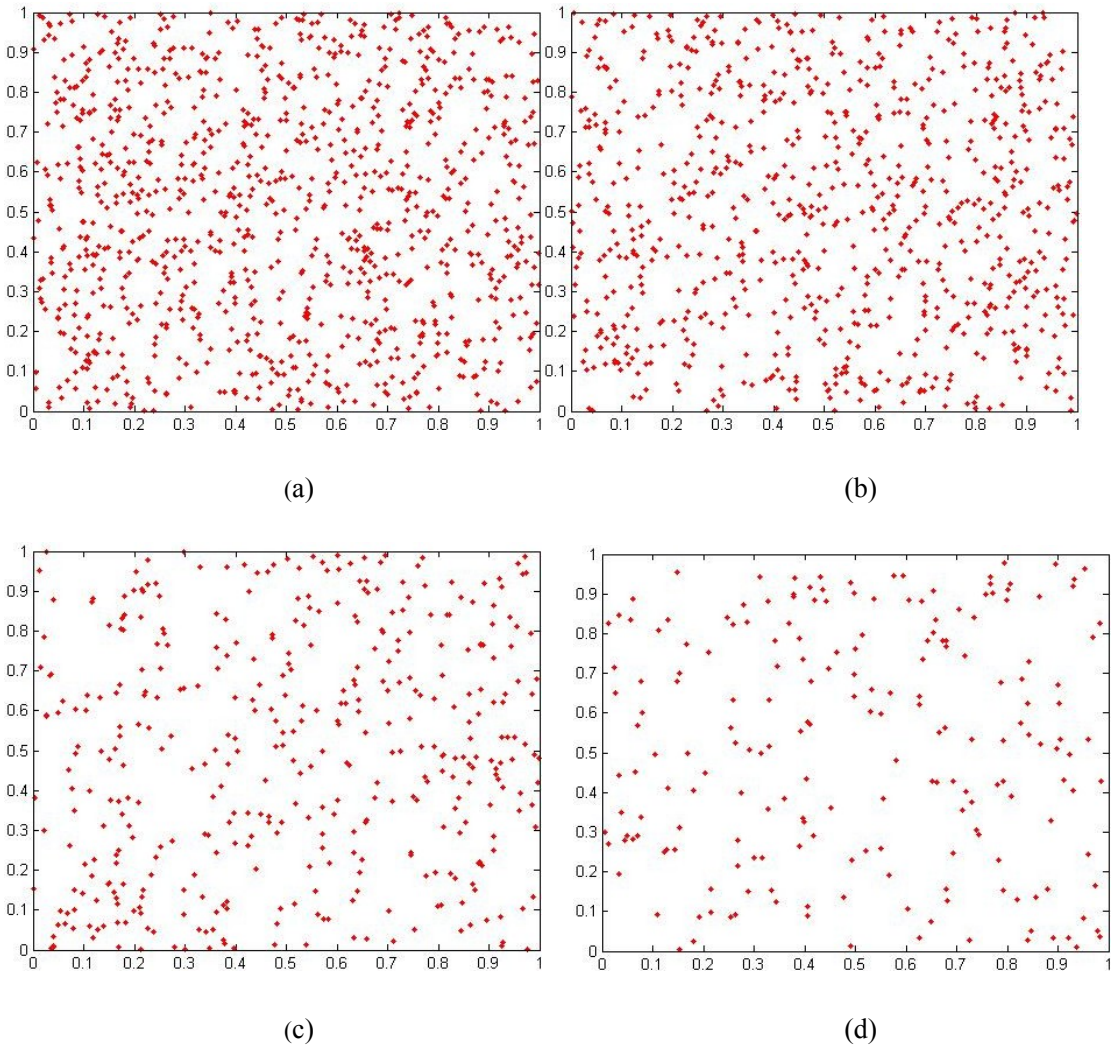
Fig.4.12 Average number of clusters in varying initial conditions. a) Initially regularly distributed data points; b) compare to initially randomly distributed data points.

According to Table 4.1, Table 4.2 and Fig.4.12, the initial condition slightly affects steady state. During the simulation, communications happen randomly, and therefore results can be different as the system evolves as a random process. For example, in the regularly distributed data condition, when the bound of confidence equals 0.11, the average of the final number of clusters is 20. But for each individual case, the final number of clusters range from 17 to 23.

In brief, the initial conditions are not the main factor that can affect the result of the number of clusters.

4.2.3 Initial Condition: Different Number of Data Points

Here, we are applying the same method and process to deal with different number of data points. The numbers of data points for different cases are 1000, 800, 400 and 200.



(a) (b)

(c) (d)

Fig.4.13 Initial condition of randomly distributed data: (a) the number of data points is 1000 ;(b) the number of data points is 800; (c) the number of data points is 400; (d) the number of data points is 200.

We run the simulation ten times and get the average number of clusters under different scenarios. The results show in Table 4.3 and Fig. 4.14.

Table 4.3 Average number of clusters with different data points from 1000 to 200 (ten runs)
(# means the number of data points.)

# \ ε	0.10	0.11	0.12	0.13	0.14	0.15	0.20	0.25	0.30	0.35	0.40
1000	23	20	17	15	12	10	5	4	2	1	1
800	24	20	17	14	12	10	5	4	2	1	1
400	26	20	19	15	12	10	6	4	2	1	1
200	27	20	18	14	13	12	6	4	3	2	1

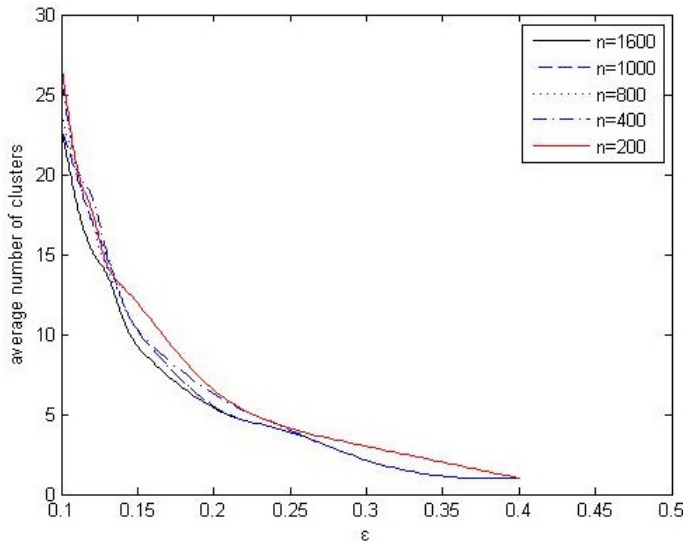


Fig.4.14 Average number of clusters in varying number of data points.

According to Table 4.1, Table 4.3 and Fig. 4.14, the results for those data points, whose number is from 400 to 1600, are virtually the same when epsilon is larger than 0.2. Therefore, the factor that seems to matter is epsilon, as curves for small epsilon are relatively different, whereas they tend to overlap into two different groups for

larger epsilon. On the other hand, the number of data points is not the main factor that affects the result of the final number of clusters.

4.2.4 Study on Time Varying Opinion Space

This section is dedicated to the simulation of opinion dynamics with DW model in a time varying opinion space. The influence of velocity of shrinking the opinion space on the average number of clusters is studied. This process is also under different bound of confidence. Additionally, the results will be compared to the situation under the static opinion space which means the opinion space is one-dimensional and stable. Finally, one extra scenario will be presented.

4.2.4.1 Simulation on the Time Varying Opinion Space

In this part, we study the influence of time varying opinion space on the final number of clusters. We move one side of the box representing the two dimensional opinion space until it becomes a line. The process is structured into several steps.

The initial condition is the same as that shown in Fig.4.15, which is a unit box with 1600 randomly distributed data points. Then, one side of the box (y-axis) shrinks from point (0, 0) to point (1, 0). When a point collides with the moving y-axis, it locates on the y-axis, which means its y-coordinate does not change and its x-coordinate changes to the new coordinate value of the y-axis. Finally, after all data points gather in a line, check if the system is steady state. Otherwise continue the iteration until the system

reaches steady state.

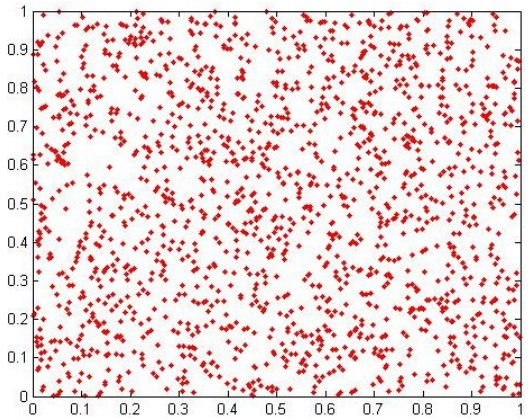
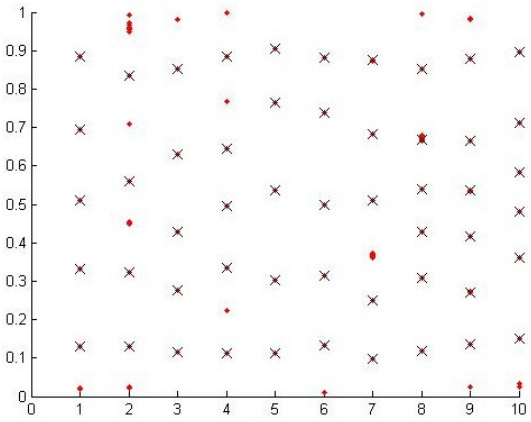


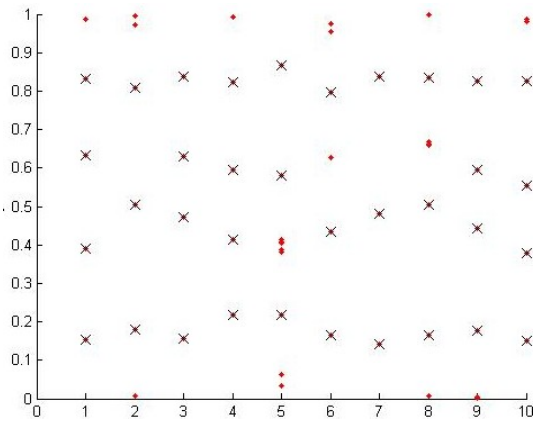
Fig.4.15 A unit box with 1600 randomly distributed points

An important parameter is the rate of iteration in shrinking the opinion space. Assuming that there will be 100 steps to move the y-axis from one side to the other, the rate is defined as the number of iterations in each step. For example, given 10 iterations as the rate in each step, the total iteration number is 1,000 until all the points gather on a line. In other words, for every 10 iteration step, the y-axis will move one step. It will get to its destination after 100 steps.

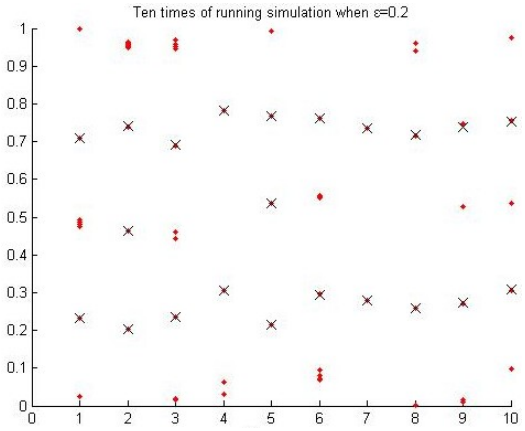
In Fig. 4.16, they are results at the rate of 10,000. In each case, with different bounds of confidence, we run the simulation ten times and finally get the average number of clusters. The following figures show individual runs with location of the centres of clusters along the y-axis in the steady state.



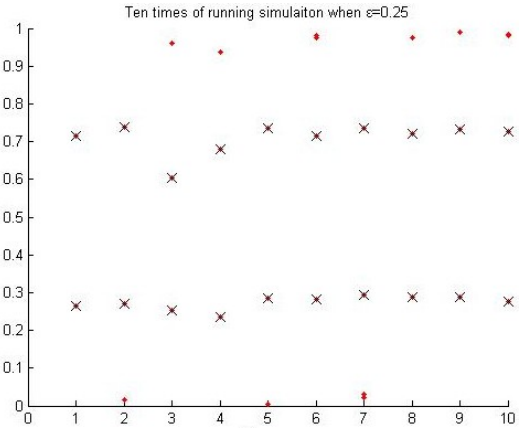
(a)



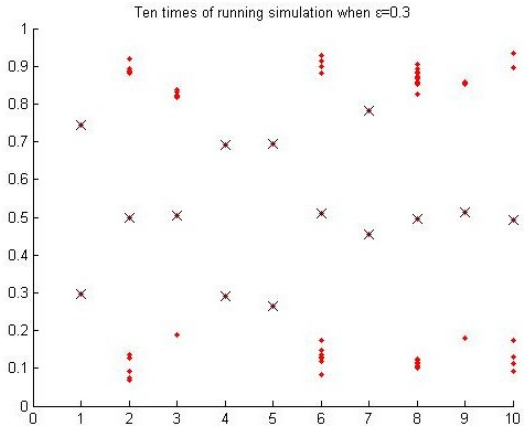
(b)


Ten times of running simulation when ε=0.2

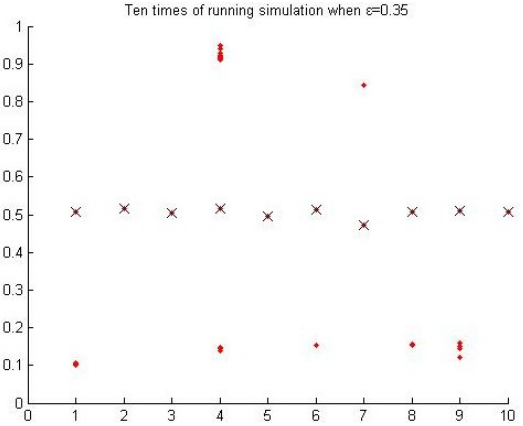
(c)


Ten times of running simulaiton when ε=0.25

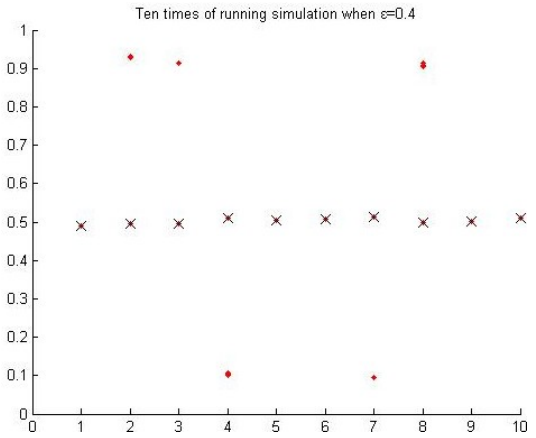
(d)


Ten times of running simulation when ε=0.3

(e)


Ten times of running simulation when ε=0.35

(f)



(g)

Fig.4.16 Simulation ten times at rate of 10,000 iterations when (a) ε equals 0.1. (b) ε equals 0.15.(c) ε equals 0.2.(d) ε equals 0.25.(e) ε equals 0.3.(f) ε equals 0.35.(g) ε equals 0.4. (The horizontal axis is the number of runs, and the vertical axis is the opinion space)

Table 4.4 The average number of clusters related to the bound of confidence at rate of 10,000 iterations per step.

ε	0.10	0.15	0.20	0.25	0.30	0.35	0.40
Number of clusters	5	4	2	2	1	1	1

Results in Table 4.4 summarize the average number of clusters versus the confidence bound, for a given rate of iteration, which as the bound of confidence increases, the number of clusters decreases. For large bound of confidence, the probability of exchanging opinions between two random data points must be high, which induces small number of cluster in the end. To compare the results of two-dimensional and steady opinion space (shown in Table 4.1), the number of clusters is much less.

We also run the simulation under different bounded confidence to get a bifurcation diagram as shown in Fig. 4.17. It shows the location of centres of clusters under

different bounded confidence. We can read it from right to left to find that clusters split into more clusters. The value of bounded confidence where this happens is called a bifurcation point [3].

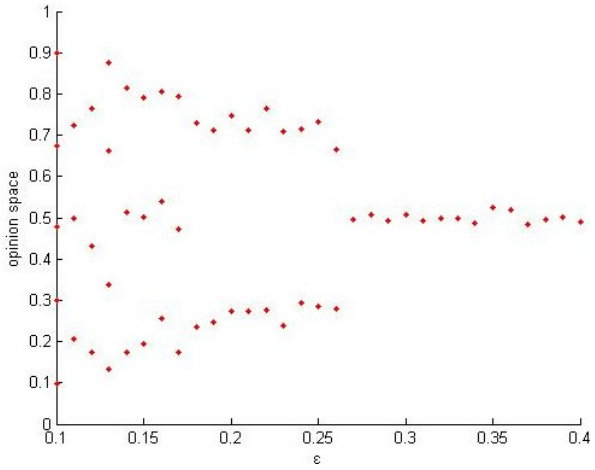
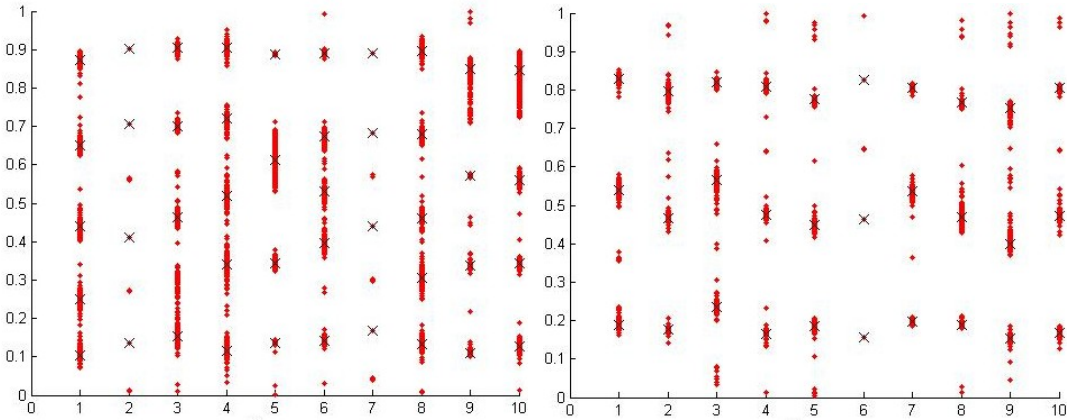
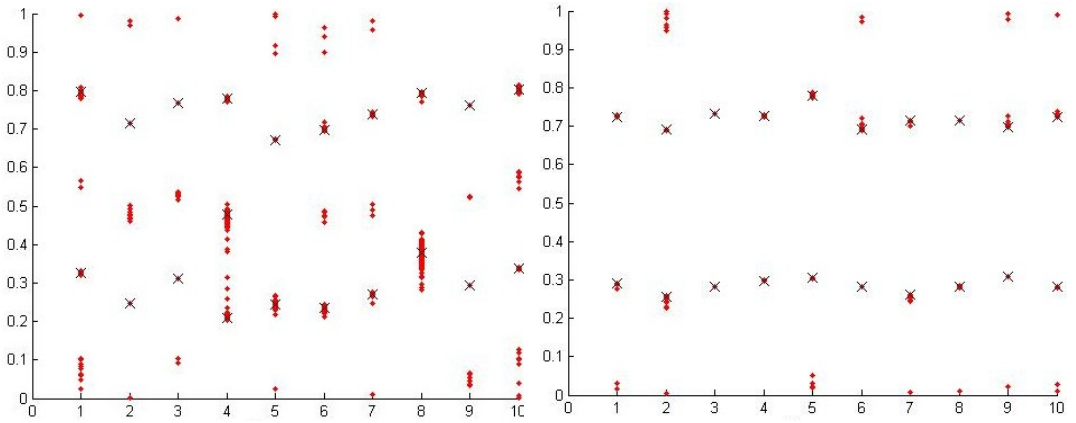


Fig.4.17 Bifurcation diagram

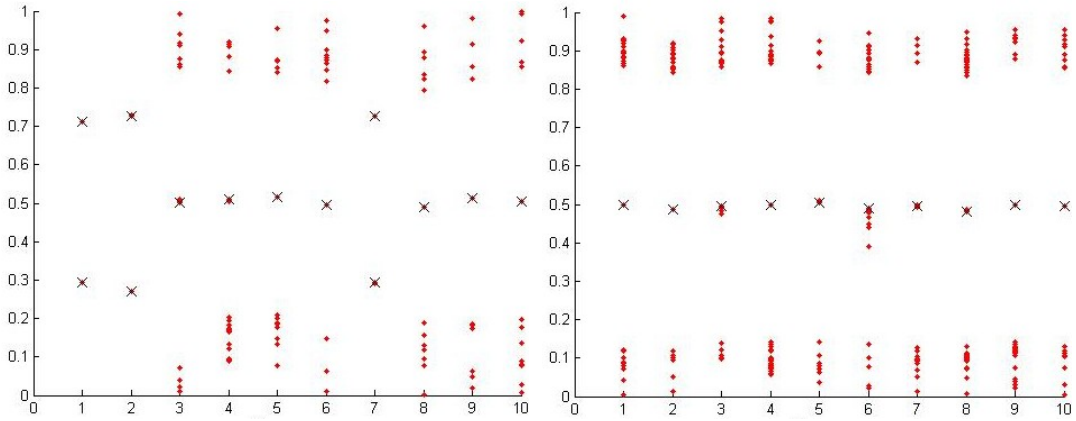
We move forward to other topics on how the results are at other rate and how the rate affects the final number of clusters. Here is an example to test the result at the rate of 1000 iterations. In each case, with different bounds of confidence, we run the simulation 10 times. The following figures show individual runs with location of the centres of clusters along the y-axis in the steady state.



(a) (b)



(c) (d)



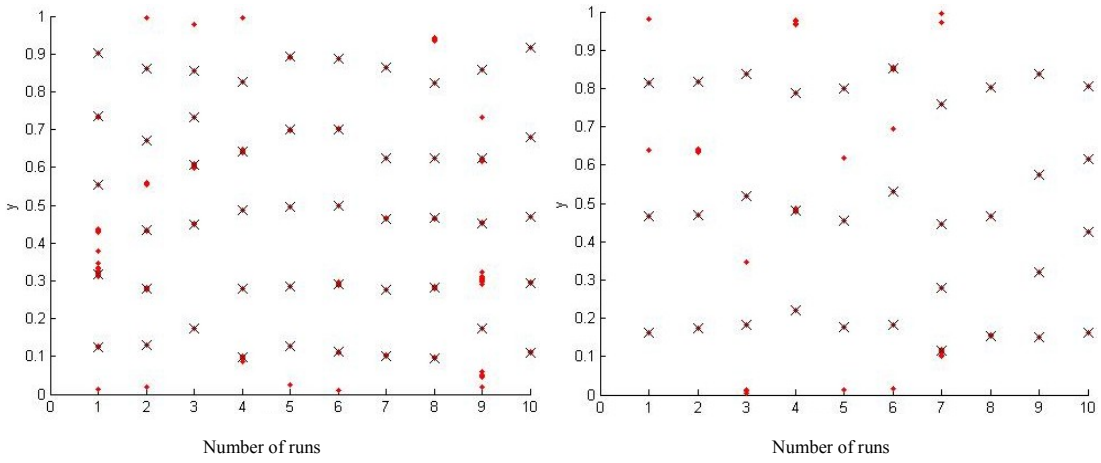
(e) (f)

Fig.4.18 Simulation ten times at rate of 1,000 iterations when (a) ε equals 0.1; (b) ε equals 0.15; (c) ε equals 0.2; (d) ε equals 0.2.5; (e) ε equals 0.3; (f) ε equals 0.35. (The horizontal axis is the number of runs, and the vertical axis is the opinion space)

Table 4.5 The average number of clusters related to the bound of at the rate of 1,000 iterations per step.

ε	0.10	0.15	0.20	0.25	0.30	0.35
Number of clusters	4	3	2	2	1	1

Therefore, according to Fig. 4.18, Table 4.4 and Table 4.5, when the bound of confidence is 0.1 or 0.15, as the rate decreases from 10000 to 1000, the average number of clusters decreases. However, for large values of the confidence bound, the rate does not affect the result of the final number of clusters. To get a more comprehensive result, we run several tests at different rates when the bound of confidence equals 0.1 and 0.15.



(a) (b)

Fig.4.19 Simulation ten times at rate of 5,000 iterations: when a) ε equals 0.1; b) ε equals 0.15.

Based on Fig. 4.19, we can tell that the average number of clusters at a rate of 5,000 iterations is 4.9. According to Fig. 4.18 (a) and Fig. 4.16 (a), the number at rate of 1,000 is 4.4 and the one at rate of 10,000 is 5.2. Therefore, we can get a conclusion shown in Fig. 4.20 that as the value of rate of iterations in each step increases, the average number of clusters tends to increase for small confidence bounds.

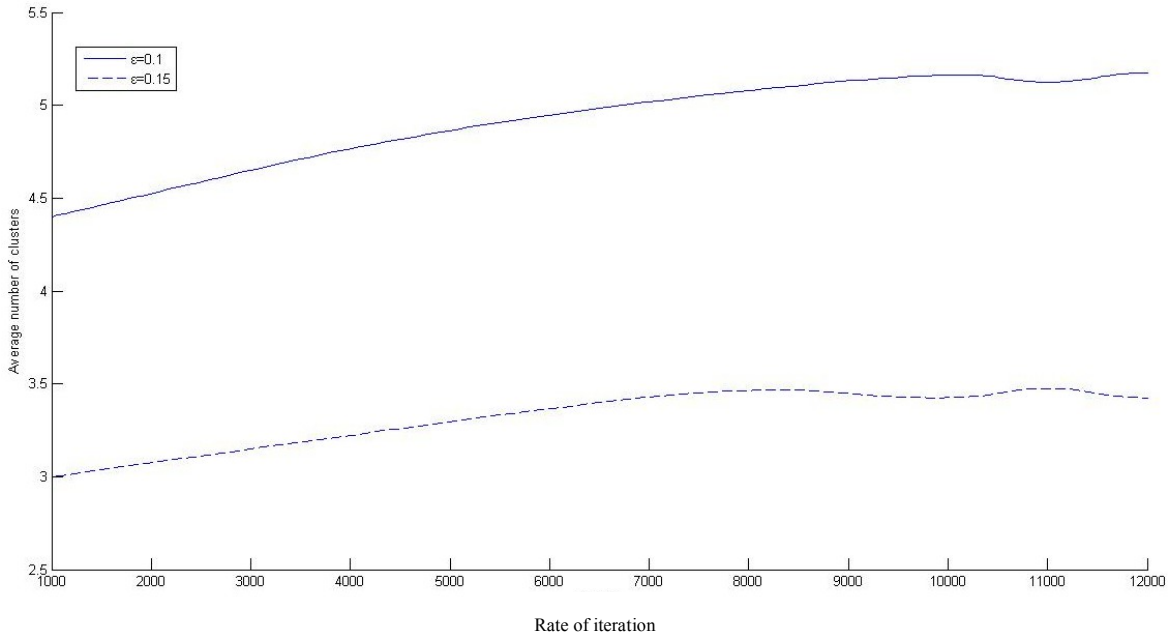


Fig.4.20 Average number of clusters at different rate of iteration under different bound of confidence

4.2.4.2 Compared to Static Opinion Space

Here we present some results showing a comparison between time-varying and static opinion space. The static opinion space is one-dimensional and the same as the final state of time-varying opinion space (from two-dimension to one-dimension). The simulation in terms of static opinion space runs ten times under the same bound of confidence.

Table 4.6 The average number of clusters related to the bound of confidence in static opinion space.

ε	0.10	0.15	0.20	0.25	0.30	0.35	0.40
Number of clusters	4	3	2	2	1	1	1

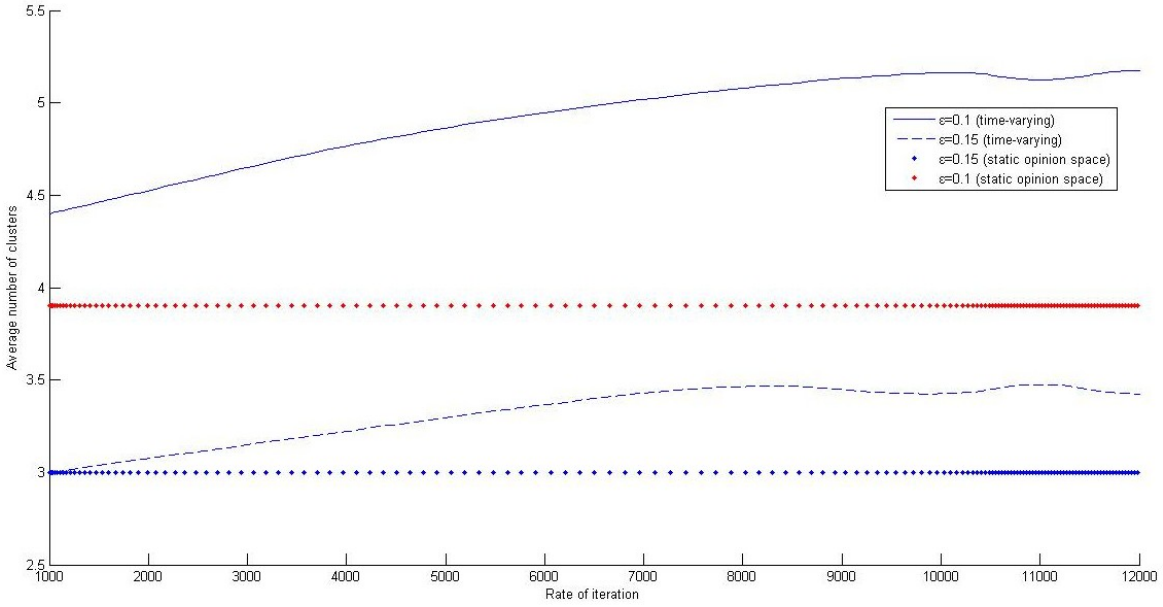


Fig.4.21 Comparison between time-varying and static opinion space

A comparison of average number of clusters in time-varying and static opinion space is presented in Table 4.6 and Fig. 4.21. We can tell that when the bound of confidence is 0.1 and 0.15, the final average number of clusters for time-varying space is larger than the one for static opinion space under the same bound of confidence. As the rate of iteration in each step increases, the gap between time-varying and static opinion space increases. This is an interesting result, as it shows a sort of path dependent behavior, in the sense that one can modify the steady state associated to a one dimensional opinion space by introducing another opinion that within the model simulated here allows agents to communicate. The most immediate implication is that it is possible to manipulate the distribution of opinions about a specific topic by allowing subjects also to talk about eventually unrelated topics, even in the extreme case when the opinion about the unrelated topics shrinks to complete agreement.

4.2.4.3 Additional Simulation Result

In this section, a special scenario related to the time-varying opinion space is studied. It is allowing convergence in each step for shrinking the opinion space. We are going to study that how it affects the final opinion distribution.

For this scenario, the system will reach convergence in each shrinking step. It means that the rate of iteration is different and large in each shrinking step. Besides, other initial conditions are the same as other simulations in section 4.2.4.1.

Table 4.7 The average number of clusters related to the bound of confidence when the convergence happened in every step of shrinking

ε	0.10	0.15	0.20	0.25	0.30	0.35	0.40
Number of clusters	5	4	2	2	1	1	1

Table 4.7 indicates that there is no difference compared to the situation at rate of 10,000 iterations (Table 4.4). In fact, when the rate of iteration is above 10,000, iterate after several steps even under the small bound of confidence, all the data points have already formed several highly overlapped groups. Allowing the system convergence in each step just makes groups more overlapped in the process of shrinking, which has little influence on the final number of clusters.

5 Conclusion

5.1 Summary

The purpose of our project was to study a particular type of opinion dynamics whose opinion space changes from two dimensions to one dimension. We choose the DW model which can be used in both two-dimensional and one-dimensional opinion dynamics. Then, to analysis the result of opinion dynamics, a clustering method should be chosen properly. First, four different clustering methods were tested in MATLAB and three of them can be used well with the DW model. Second, it is crucial to determine the steady state of the system and estimate the corresponding number of clusters. We combined Fuzzy c-means and subtractive clustering methods.

Simulations for testing how initial conditions and parameters affect the results were carried out in MATLAB. In the two-dimensional case with the DW model, initial conditions including randomly and regularly distributed data points, varied number of data points do not influence the final number of clusters. The only thing that affects the result is the bound of confidence. Additionally, in the simulation of moving one side of the box which is turning two-dimensional case to one-dimensional one, the overall trend is the same as the two-dimensional case: as the bound of confidence increases, the number of clusters decreases. On the other hand, we also find that as the rate of iteration in each shrinking step goes up, the average number of clusters tends

to decrease when the bound of confidence is less than 0.15. Convergence iterations are also studied in appendix. It depends on the amount number of agents, the number of final clusters, the size and degree of concentration of the final clusters, and the bound of confidence. Simulations demonstrate that as the bound of confidence increases, convergence iterations falls except for those overlapped cases because the degree of overlapping has a considerable influence on convergence iterations.

5.2 Research Contributions

There is a contribution to the field of opinion dynamics, which is to gain insights into cases of shrinking opinion space. While opinion space is reducing, it actually adds another variable which is the rate of iteration of shrinking to the process of opinion dynamics.

5.3 Future Work

There is still room to improve the clustering technique and extend study areas of time varying opinion space. Firstly, although FCM-subtractive clustering is applied quite well in the DW model, it cannot be assumed to be suitable for other models of opinion dynamics. This model is not complicated, and the key standard for estimating the steady state of clusters is very common, so the next step is to make the clustering method better to satisfy more complex models and standards of estimating clusters.

Secondly, in terms of dynamic opinion space, to make the study of dynamic opinion space widely, the next step is to study extended opinion space.

References

[1] J. Lorenz, "Fostering consensus in multidimensional continuous opinion dynamics under bounded confidence," presented at conf. Potentials of complexity Science for Business, Governments, and the Media, Budapest, Aug. 4, 2006

[2] W. Ren, R. W. Beard, "Distributed Consensus in Multi-vehicle cooperative control, " in Springer, Communications and Control Engineering, 2008, pp. 3

[3] M.A. Montes de Oca. (2010) Opinion Dynamics for Decentralized Decision-making in a Robot Swarm. Ants 2010, Lncs 6234, pp.251-262

[4] J. Lorenz. "Continuous opinion dynamics of multidimensional allocation problems under bounded confidence: More dimensions lead to better chances for consensus," presented at conf. Connectionist Approaches in Economics and Management – ASCEG, Aix-en-Provence, France, Nov. 17, 2005

[5] E. Gorbatikov. (2013, Oct.) Mathematical model of opinion dynamics in social groups. *Mediterranean Journal of Social Sciences.* 4(10), pp. 380-387. doi**:** 10.5901/mjss.2013.v4n10p380

[6] J. Lorenz. (2007, Dec.) Continuous opinion dynamics under bounded confidence: A survey. *International Journal of Modern Physics C. 18(12),* pp. 1819-1838. doi: 10.1142/S0129183107011789

[7] A. C. R. Martins. (2008). Continuous opinions and discrete actions in opinion dynamics problems. *International Journal of Modern Physics C. 19(4),* pp. 617-624.

[8] K. Sznajd-Weron. (2008, Feb.). Sznajd model and its application. *Econo and sociophysics. 36(8),* pp. 2537-2547.

[9] G. Zaklan. (2009, Jan.). Analysing tax evasion dynamics via the Ising model. *J Econ Interact Coord. 4,* pp. 1-14. doi: 10.1007/s11403-008-0043-5

[10] D. Stauffer. (2000, Sep.). Generalization to square lattice of Sznajd scociophysics model. *International Journal of Modern Physics C.* 11(6), pp. 1239-1245.

[11] Y. Zhao, "The roles of environmental noises and opinion leaders in emergency," presented at 9th International Conference on Active Media Technology, AMT 2013; Maebashi; Japan; October 29-31, 2013

[12] J. Lorenz. (2009, Nov.). Heterogeneous bounds of confidence: meet, discuss and find consensus. *Wiley InterScience. 15(4),* pp. 43-52.

[13] A. Mirtabatabaei. (2012). Opinion dynamics in heterogeneous networks: convergence conjectures and theorems. *Society for Industrial and Applied Mathematics. 50(5*), pp. 2763-2785.

[14] J. Zhang, "Convergence Analysis of Heterogeneous Deffuant-Weisbuch model," presented at the 31st Chinese Control Conference, Hefei, China. July 25-27, 2012

[15] A. Pluchino. (2006, Sep.) Opinion dynamics and synchronization in a network of scientific collaborations. *Physica A. 372,* pp. 316-325. doi:10.1016/j.physa.2006.08.016

[16] Z. Liu. (2014, May) On the control of opinion dynamics in social networks. *Physica A.* [online]. *409,* pp. 183-198. Available: http://dx.doi.org/10.1016/j.physa.2014.04.037

[17] P.M. Yonta. (2009, Mar.) Opinion dynamics using majority functions. *Mathematical Social Sciences*. 57(2), pp. 223-224. doi**:** 10.1016/j.mathsocsci.2008.12.006

[18] E. Ben-Naim. (2005, Mar.) Opinion dynamics: rise and fall of political parties. *Europhysics Letters.* 69(5), pp.671-677. doi: 10.1209/epl/i2004-10421-1

[19] M.A.M. deOca. (2011, Dec.) Majority-rule opinion dynamics with differential latency: A mechanism for self-organized collective decision-making. Swarm Intelligence. 5(3), pp. 305-327. doi**:** 10.1007/s11721-011-0062-z

[20] M. Pineda. (2009) Noisy continuous-opinion dynamics. *Journal of Statistical Mechanics: Theory and Experiment.* 2009(8), doi: 10.1088/1742-5468/2009/08/P08001

[21] L.P. Chi. (2011, Dec.) Binary opinion dynamics with noise on random networks. *Chinese Science Bulletin. 56(34),* pp. 3630-3632. doi: 10.1007/s11434-011-4751-1

[22] M. Pineda. (2014, Nov.) Mass media and repulsive interactions in continuous opinion dynamic. *Physica A. 420(2015),* pp. 73-84

[23] V.D. Blondel. (2009) On Krause's multi-agent consensus model with state-dependent connectivity. *IEEE Trans. Automat. Control. 54 (11),* pp. 2585–2597

[24] J. Lorenz. (2005) A stabilization theorem for dynamics of continuous opinions. *Physica A, 355 (2005),* pp. 217–223

[25] A. Nedic, "Multi-dimensional Hegselmann–Krause dynamics", presented at 51st IEEE Conference on Decision and Control, pp. 68–73, 2012.

[26] A. Mirtabatabaei, "On opinion dynamics in heterogeneous networks", presented at 2011 American Control Conference, pp. 2807–2812, 2011.

[27] I.-C. Morarescu, "A model of opinion dynamics for community detection in graphs", presented at 2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems, NecSys'10, Annecy, France, September 13-14, 2010

[28] I.-C. Morarescu. (2011, Aug.) Opinion dynamics with decaying confidence: Application to community detection in graphs. *IEEE Transactions on Automatic Control, 56(8),* pp. 1862-1873. doi: 10.1109/TAC.2010.2095315

[29] M. Halkidi. (2002, Jun.) Cluster Validity Methods: Part1. *SIFMOD Record*. 31(2), pp.40-45.

[30] A.K. Jain. (2010) Data clustering: 50 years beyond k-means. *Pattern Recognition Letters. 31(2010),* pp. 651-666. doi:10.1016/j.patrec.2009.09.011

[31] H.B. Zhou, "Automatic method for determining cluster number based on silhouette coefficient", presented at 3rd International Conference on Intelligent Materials and Mechanical Engineering, Guangzhou; China; May 24-25, 2014, pp. 227-230

[32] P.J. Rousseeuw. (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics 20,* pp. 53-65. doi: 10.1016/0377-0427(87)90125-7

[33] D. Zhu. (2013) Determining the number of clusters by a bayesian approach. *International Journal of Applied Mathematics and Statistics. 47(17),* pp. 10-14.

[34] M. Yan. (2007, Dec.) Determining the number of clusters using the weighted gap statistic. *Biometrics. 63(4),* pp. 1031-1037. doi: 10.1111/j.1541-0420.2007.00784.x

[35] J.M. Chen, "A method for determining the number of clusters based on graph theory", presented at 7th International Conference on Fuzzy Systems and Knowledge Discovery, Yantai, Shandong, China, August 10-12, 2010, pp. 2706-2710

[36] J.-S. Lee. (2013, Jan.) A meta-learning approach for determining the number of clusters with consideration of nearest neighbors. *Information Sciences. 232(2013),* pp. 208-224. doi: 10.1016/j.ins.2012.12.033

[37] H. Sun. (2004, Oct.) FCM-based model selection algorithms for determining the number of clusters. *Pattern Recognition. 37(10),* pp. 2027-2037.

[38] Z. Qin. (2013, Aug.) A method for determining optimal number of clusters based on K-means algorithm. *Journal of Computational Information Systems. 9(15*), pp. 6123-6130.

[39] T. Kanungo. (2002, Jul.) An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence. 24(7),* pp. 881-892.

[40] X.Y. Wang. A comparison of fuzzy and non-fuzzy clustering techniques in cancer diagnosis. [Online]. Available: http://ima.ac.uk/papers/wang2005.pdf

[41] T. Liu, "Clustering billions of images with large scale nearest neighbor search", presented at 7th IEEE Workshop on Applications of Computer Vision, TX, United States; February 21-22, 2007

[42] O.J. Oyelade. (2010) Application of k-means clustering algorithm for prediction of students' academic performance. *International Journal of Computer Science and Information Security. 7(1),* pp. 292-295

[43] A. Saurabh, "Wireless sensor network based adaptive landmine detection algorithm", presented at 2011 3rd International Conference on Electronics Computer Technology, Kanyakumari, India. April 8-10, 2011, pp. 220-224

[44] J. B. MacQueen (1967). "Some Methods for classification and Analysis of Multivariate Observations". Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1. University of California Press. pp. 281–297

[45] K. Hammouda, A comparative study of data clustering techniques. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.126.3224&rep=rep1&type=pdf

[46] Gaussian mixture models, Matlab R2014b Documentation [Online]. Available: http://www.mathworks.com/help/stats/gaussian-mixture-models.html

[47] J.C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms." Kluwer Academic Publishers Norwell, MA, USA, 1981

[48] K-means separates data into Voronoi-cells [Online]. Available: http://datab.us/i/cluster%20analysis

[49] D. Reynolds, Gaussian mixture models. [Online]. Available: http://www.ll.mit.edu/mission/cybersec/publications/publication-files/full_papers/0802_Reynolds_Biometrics-GMM.pdf

[50] J.C. Bezdek (1984). FCM : the fuzzy c-means clustering algorithm. *Computers & Geosciences. 10(2-3),* pp.191-203

[51] W.Y. Liu, "Study on combing subtractive clustering with fuzzy c-means clustering", presented at 2003 2nd International Conference on Machine Learning and Cybernetics, Xi'an, China. November 2-5, 2003, pp. 2659-2662

[52] Q. Yang, "An initialization method for fuzzy c-means algorithm using subtractive clustering", presented at 2010 3rd International Conference on Intelligent Networks and Intelligent Systems. November 1-3, 2010, pp. 393-396

[53] D.H. Nam, "Material processing of ADI data using neuro fuzzy system", presented at 19th International Conference on Computer Applications in Industry and Engineering, Las Vegas, NV, United States. November 13-15, 2006, pp. 151-156

[54] K. Z, "An improved FCM algorithm for color image segmentation", presented at 3rd International Conference on Innovative Computing Information and Control, Dalian, Liaoling, China. June 18-20, 2008, pp. 200-203

[55] k-means clustering, Matlab R2014b Documentation, [Online]. Available: http://www.mathworks.com/help/stats/k-means-clustering.html

[56] Fuzzy Clustering, Matlab R2014b Documentation [Online]. Available: http://www.mathworks.com/help/fuzzy/fuzzy-clustering.html#FP2434

[57] Subtractive clustering, Matlab R2014b Documentation [Online]. Available : http://www.mathworks.com/help/fuzzy/subclust.html

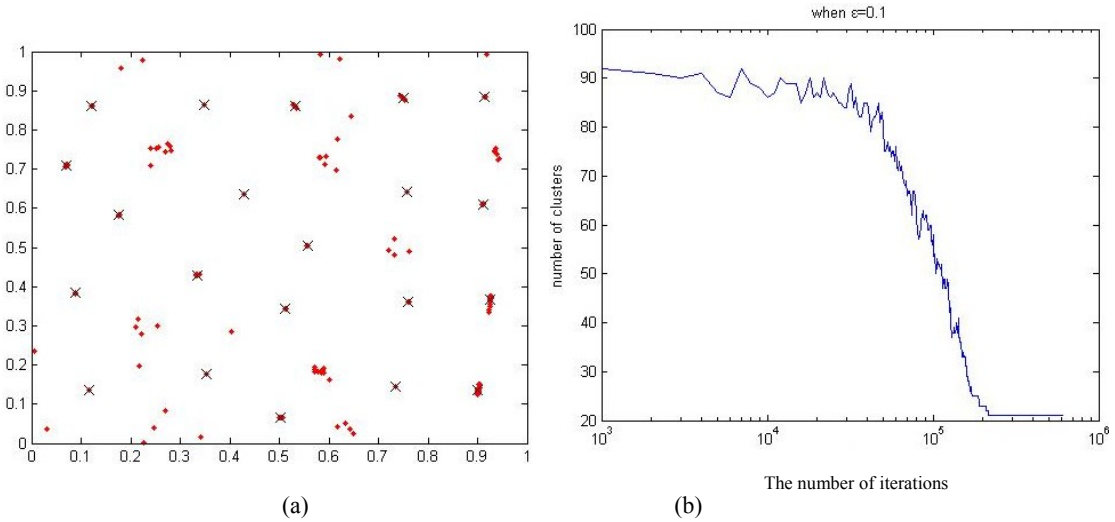
[58] A. Fujita. (2013) A non-parametric method to estimate the number of clusters. *Computational Statistics and Data Analysis. 73(2014),* pp. 27-39

Appendix A

Convergence Iterations Analysis

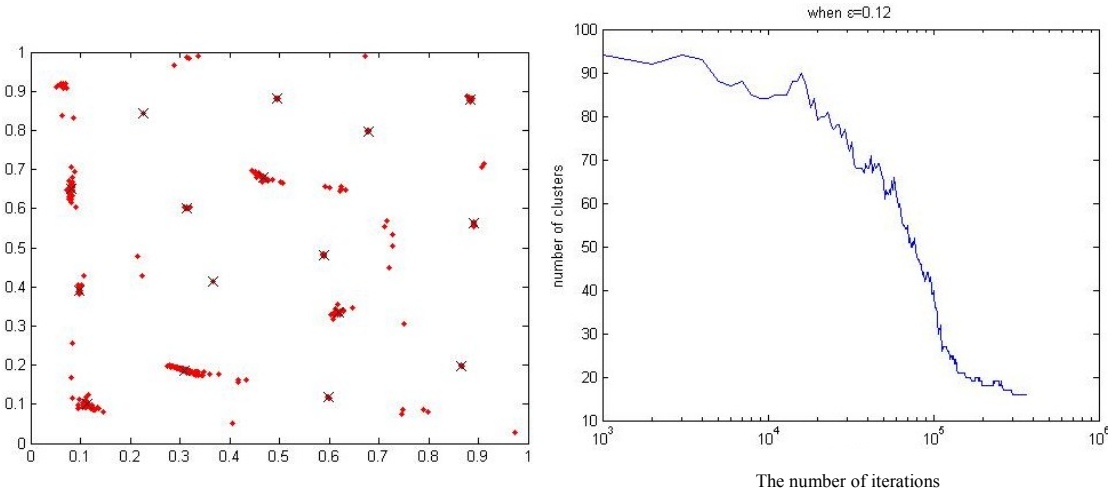
In the DW model, just two agents communicate and make decisions at each step. Intuitively, as the value of bound of confidence increases, the ratio of successful communication (they change their opinions) versus number of iterations increases, which induces a decrease of convergence time of the system. In fact, the convergence time depends on the number of agents, the number of final clusters, degree of overlapping data for each cluster, and the bound of confidence. Here, we are giving some examples to study the convergence iterations issue.

Taking the same initial condition as in Section 4.2.1, we focus on the relationship among convergence iterations, the bound of confidence and the final number of clusters.



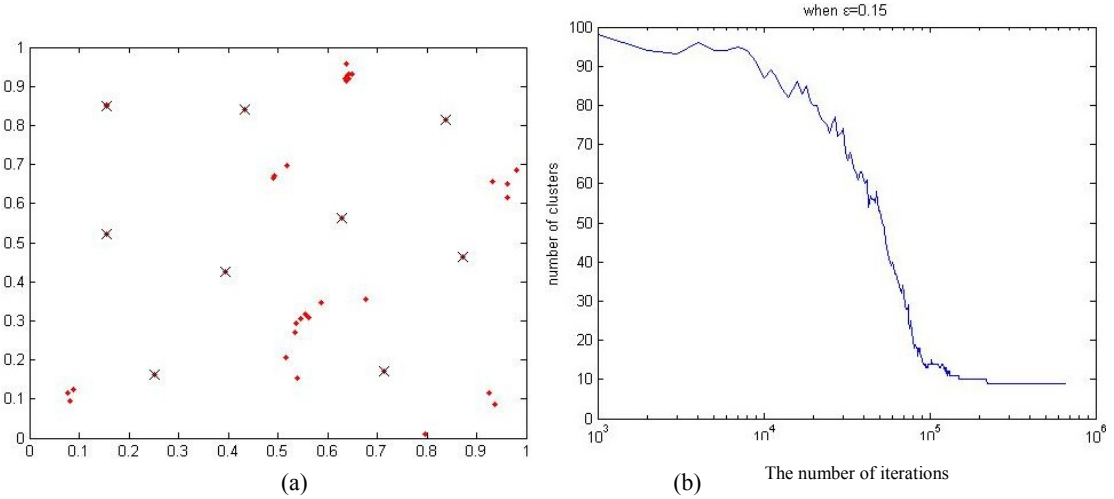
(a) (b)

Fig.A1 Convergence iterations when ε equals 0.1. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters



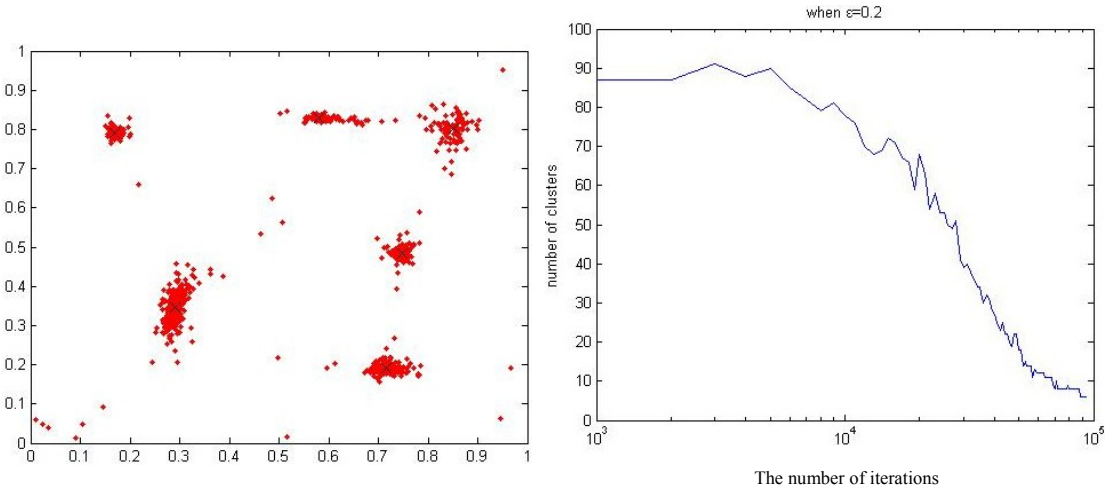
(a) (b)

Fig.A2 Convergence iterations when ε equals 0.12. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters



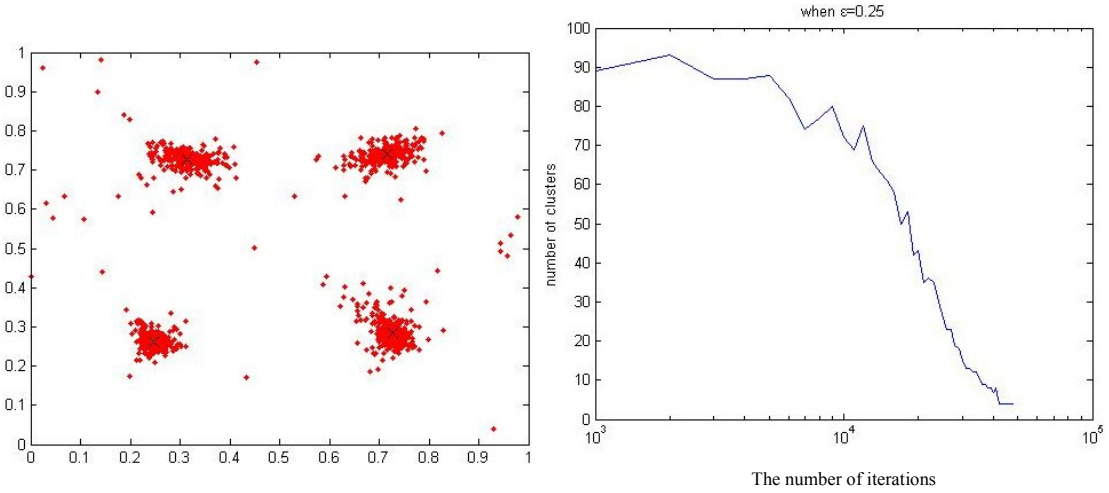
(a) (b)

Fig.A3 Convergence iterations when ε equals 0.15. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters



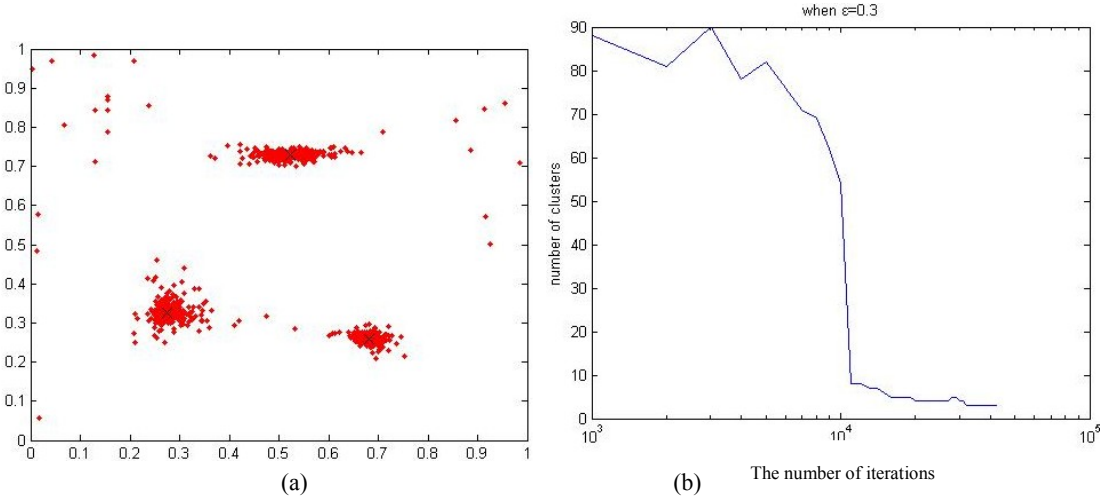
(a) (b)

Fig.A4. Convergence iterations when ε equals 0.2. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters



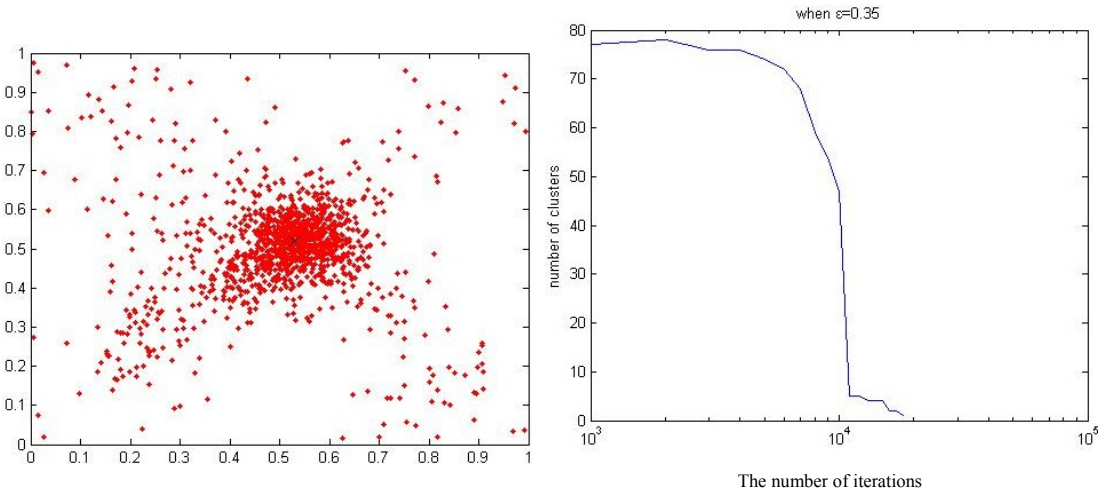
(a) (b)

Fig.A5 Convergence iterations when ε equals 0.25. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters



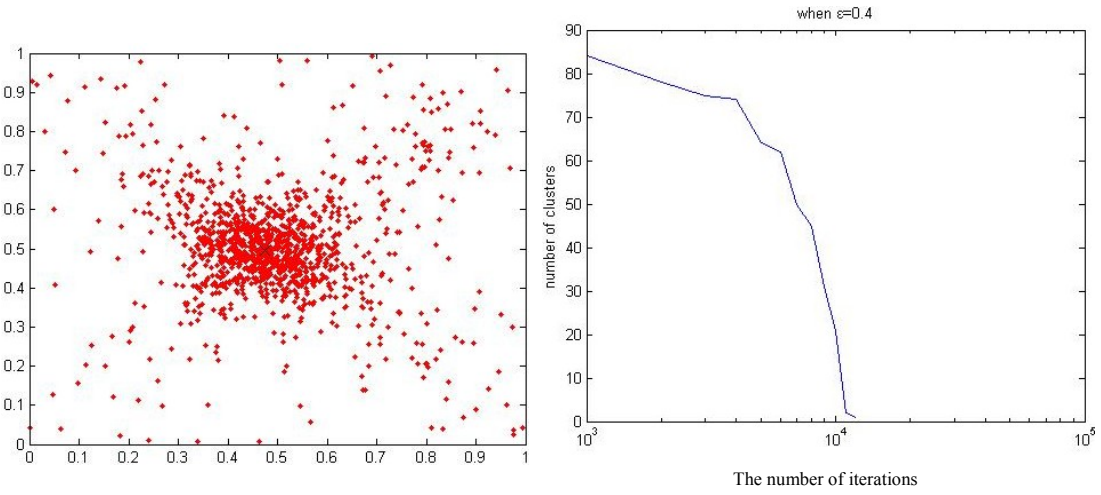
(a) (b)

Fig.A6 Convergence iterations when ε equals 0.3. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters



(a) (b)

Fig.A7 Convergence iterations when ε equals 0.35. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters



(a) (b)

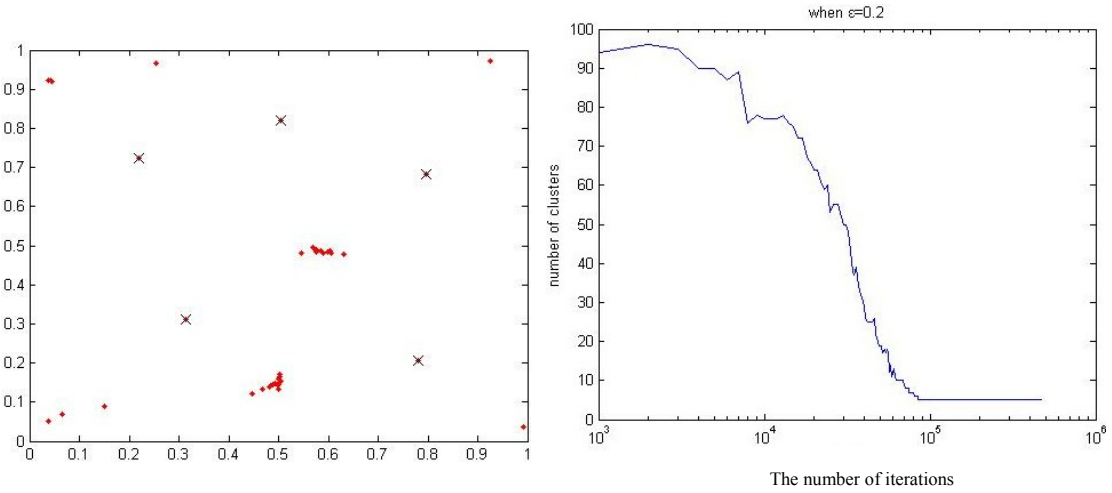
Fig.A8 Convergence iterations when ε equals 0.4. (a) the final convergence state. (b) the relation between simulation iterations and number of clusters

Table A1 Convergence iterations under different bound of confidence

ε	0.1	0.12	0.15	0.2	0.25	0.3	0.35	0.4
Convergence iterations	602000	365000	660000	93000	48000	42000	18000	12000
Number of clusters	21	16	9	6	4	3	1	1

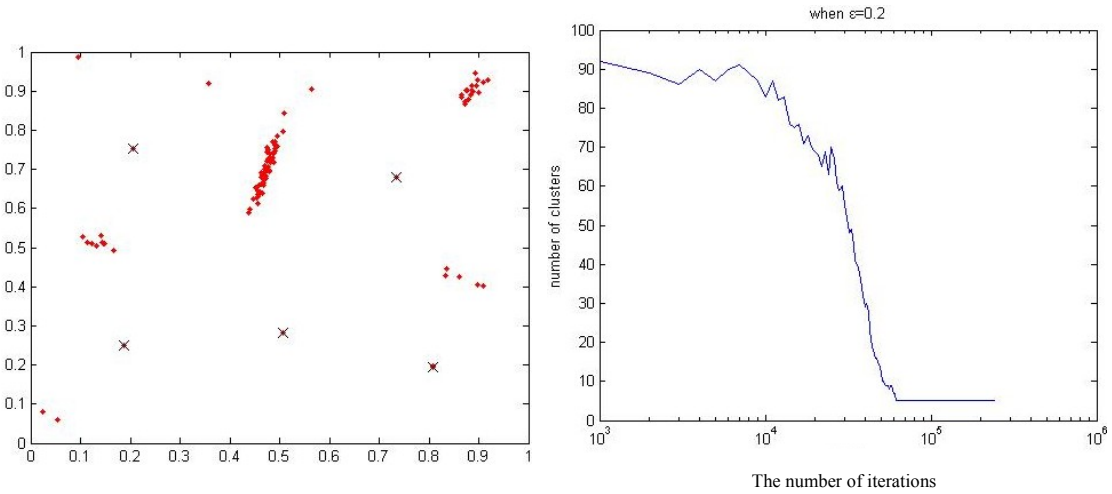
According to Table A1, convergence iterations decrease as the bound of confidence increases. However, there is an exceptional case when ε equals 0.15. Its convergence iterations are larger than the one when ε equals 0.1. To compare Fig. A1 with Fig.A3, the data points of Fig.A3 (a) is more overlapped. It means that it takes a longer time to communicate among agents in the same cluster.

In the following, more examples show that the relation between convergence iterations and the number of clusters include degree of data overlapping. Run simulation for five more times under the bound of confidence 0.2.



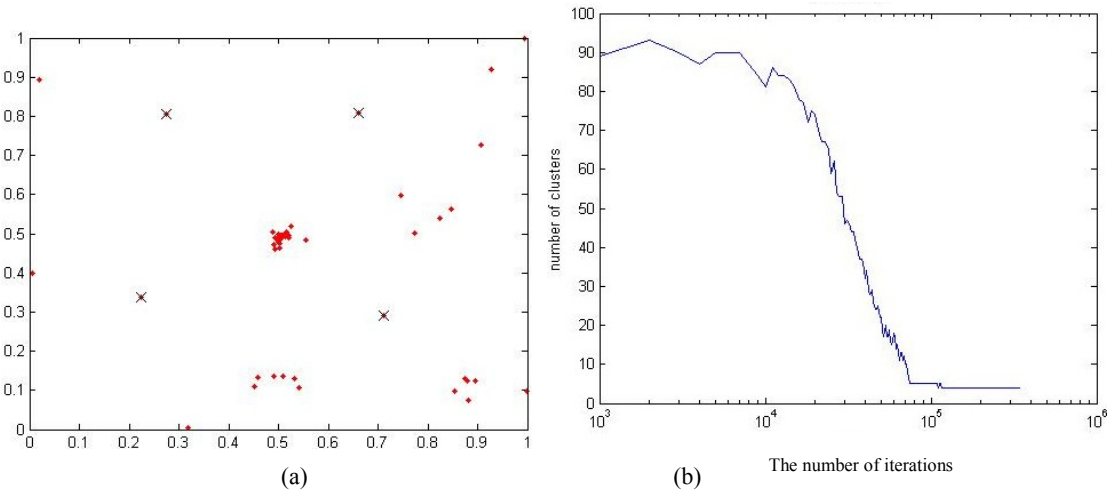
(a) (b)

Fig.A9 The 2nd time simulation when ε equals 0.2. (a) final convergence state. (b) the relation between simulation iterations and number of clusters



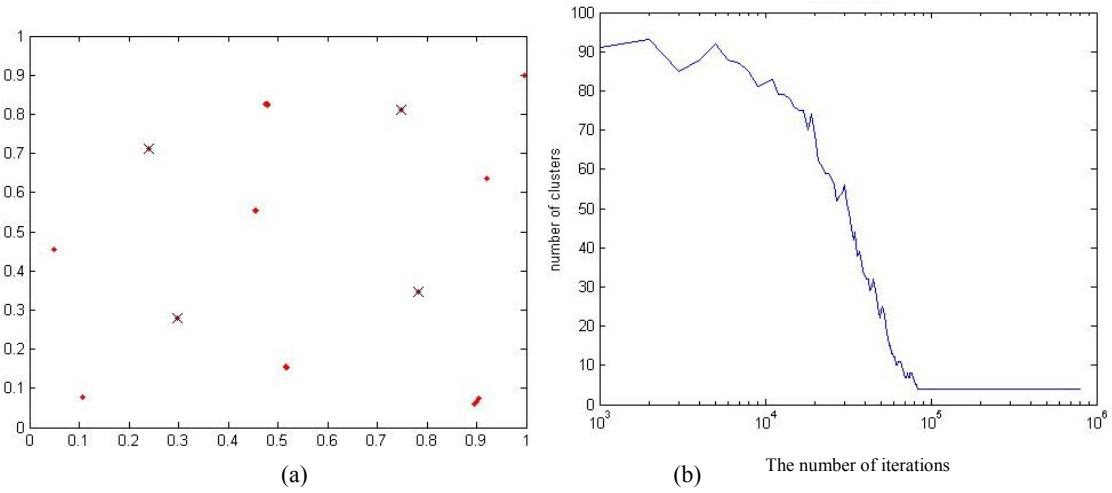
(a) (b)

Fig.A10 The 3rd simulation time when ε equals 0.2. (a) final convergence state. (b) the relation between simulation iterations and number of clusters



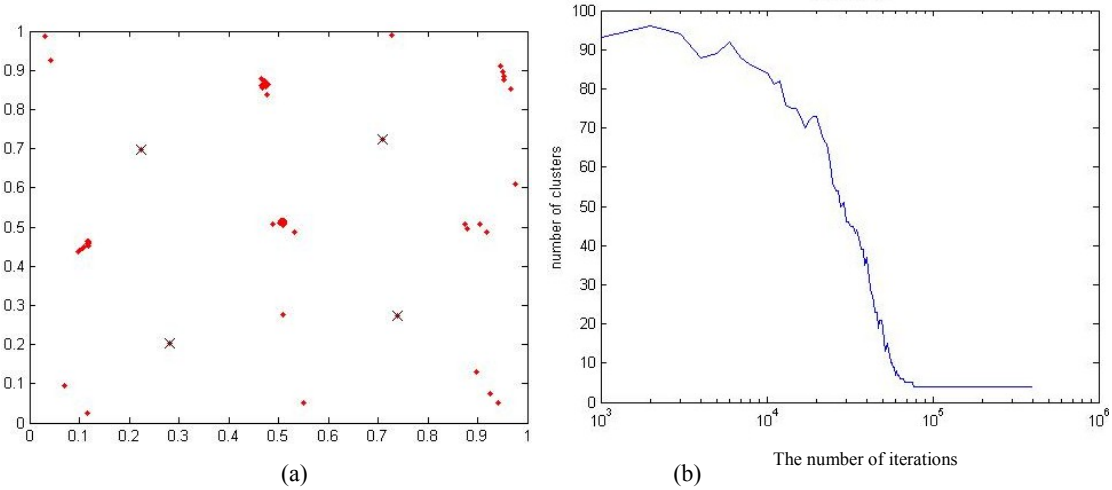
(a) (b)

Fig.A11 The 4th simulation time when ε equals 0.2. (a) final convergence state. (b) the relation between simulation iterations and number of clusters



(a) (b)

Fig.A12 Simulation the 5th time when ε equals 0.2. (a) final convergence state. (b) the relation between simulation iterations and number of clusters



(a) (b)

Fig.A13 The 6th simulation time when ε equals 0.2. (a) final convergence state. (b) the relation between simulation iterations and number of clusters

Table A2 Six times of simulation when ε equals 0.2

	1st	2nd	3rd	4th	5th	6th
Convergence iterations	93000	467000	242000	341000	795000	397000
Number of clusters	6	5	5	4	4	4

According to Table A2, the convergence iterations vary from 93,000 to 397,000, even under the same bound of confidence. The convergence iterations of the 5th is even

larger than the one of 0.12 as the bound of confidence in table 4.2. It means that when two bound of confidence are very close, we cannot confirm that the less one will take more iterations to reach convergence. The degree of overlapping for data points may have considerable influence on the convergence iterations. Under the influence of overlapping, it is not difficult to understand the differences among the 4^{th}, the 5^{th}, and the 6^{th} simulations, even if they have the same final number of clusters.

Appendix B

MATLAB: Opinion Dynamics Code

B.1 Main Function

```
clear;clc;
%Assign initial conditions%;
NN=1600; % the number of agents%
a=0.4;% bound of confidence %
X0=random('Uniform',0,1,NN,2);%NN*2 matrix from 0 to1 %

%DW Model
time=1000;
X0 =DW_model( X0,NN,time,a );
%get the number of cluster
[C,S] = subclust(X0,0.1);
[m,n]=size(C); % m is the number of clusters
[centre,U,objFcn] = fcm(X0,m);

if (m>1)
%find which cluster each data belongs to
idx_cluster = idx_clust(U,NN,m);
%check distance between two points which are not belong to the same cluster
count = distance_check( NN,a,idx_cluster,X0 );
% continue calculating
while (count>0)
time = 1000;
X0 =DW_model( X0,NN,time,a );

[C,S] = subclust(X0,0.1);
[m,n]=size(C);
if (m==1)
    break
end
[centre,U,objFcn] = fcm(X0,m);

idx_cluster = idx_clust(U,NN,m);
```

```
count = distance_check( NN,a,idx_cluster,X0 );

end

end

figure (1)

plot(X0(:,1),X0(:,2),'r.','markersize',5);
hold on;
plot(C(:,1),C(:,2),'kx','markersize',10);
```

B.2 DW Model Function

```
function [ X0 ] = DW_model( X0,NN,time,a )
   for t=0:1:time
      i=randi([1 NN]);
      j=randi([1 NN]);
      mean=0.5*(X0(i,:)+X0(j,:));
         if (norm(X0(i,:)-X0(j,:)) <= a)
            X0(i,:)=mean;
            X0(j,:)=mean;
         end
   end
End
```

B.3 Check distance Function

```
function [ count ] = distance_check( NN,a,idx_cluster,X0 )
count=0;
for i=1:1:NN
      for j=i+1:NN
         if (idx_cluster(1,i)~= idx_cluster(1,j))
            if (norm(X0(i,:)-X0(j,:))<=a && norm(X0(i,:)-X0(j,:))~=0)
                  count=count+1;
                  break
            end

         end
      end

end
```

B.4 Find data's belonging Function

```
function [ idx_cluster ] = idx_clust( U,NN,m )
maxU=max(U);
idx_cluster = [];
for i=1:NN
      for j=1:m
            if (U(j,i)==maxU(1,i))
                  idx_cluster=[idx_cluster,j];
            end
      end
end

end
```

B.5 GMM test

```
clear; clc;
load('20140522.mat');
idx=gmm(X0,3);
for i=1:1:1600
    if idx(i,1) > idx(i,2) &&    idx(i,1) > idx(i,3);
          dx(i,1)=1;
    end
    if idx(i,2) > idx(i,1) &&    idx(i,2) > idx(i,3);
          dx(i,1)=2;
    end
    if idx(i,3) > idx(i,1) &&    idx(i,3) > idx(i,2);
          dx(i,1)=3;
    end
end
plot(X0(dx==1,1),X0(dx==1,2),'r.','MarkerSize',12);
hold on;
plot(X0(dx==2,1),X0(dx==2,2),'b.','MarkerSize',12);
hold on;
plot(X0(dx==3,1),X0(dx==3,2),'k.','MarkerSize',12);
```

B.6 Find the number of points in an overlapping data case

```
X=X0*100;
X=round(X);
X=X/100;
% R = vpa(A,d) uses at least d significant (nonzero) digits, instead
```

```
%of the current setting of digits.
CountN=[0,0];t=1;
for i=1:NN
      CountN(t,3)=0;check=0;
      %check whether X0(i,:) exist in CountN
      for k=1:t
          if(X(i,1)==CountN(k,1)&&X(i,2)==CountN(k,2))
                check=check+1;
          end
      end
      if (check==0)
          for j=i:NN
              if(X(i,:)==X(j,:))
                 CountN(t,1)=X(i,1);
                 CountN(t,2)=X(i,2);
                 CountN(t,3)=CountN(t,3)+1;
              end
          end
        t=t+1;
      end
end
```

B.7 Shrinking opinion space

```
clear;clc;
%Assign initial conditions%
for j=1:10
NN=1600; % the number of agents%
a=0.4;% confidence bound%
X0=random('Uniform',0,1,NN,1);%NN*2 matrix from 0 to1 %

time=50000;

X0 =DW_model( X0,NN,time,a );%DW Model


%get the number of clusters
[C,S] = subclust(X0,0.3);
[m,n]=size(C); %cluster number
[centre,U,objFcn] = fcm(X0,m);

if (m>1)
%find which cluster each data belongs to
idx_cluster = idx_clust(U,NN,m);
```

```
%check distance between two points which are not belong to the same cluster
count = distance_check( NN,a,idx_cluster,X0 );
% continue calculating
while (count>0)
time = 1000;
X0 =DW_model( X0,NN,time,a );

[C,S] = subclust(X0,0.1);
[m,n]=size(C);
if (m==1)
      break
end
[centre,U,objFcn] = fcm(X0,m);

idx_cluster = idx_clust(U,NN,m);
count = distance_check( NN,a,idx_cluster,X0 );

end

end
hold on;
axis([0,10,0,1]);axis manual;hold on;
plot(j,X0(:,1),'r.','markersize',5);
hold on;
plot(j,centre(:,1),'kx','markersize',10);
end
hold on;
title('simulate ten times');xlabel('Times');ylabel('y');
```

B.8 A video of shrinking opinion space

```
%moving one side of the box
clear;clc;
%Assign initial conditions%

NN=1600; % the number of agent
a=0.2;% confidence bound%
X0=random('Uniform',0,1,NN,2);%NN*2 matrix from 0 to1 %

time=500;
j=0;
for y=0:0.01:1 %define velocity of y axis.
    X0 =DW_model( X0,NN,time,a );%DW Model
```

```
    %check data. If some data's x-coordinate is small than 0.01, then change
    %it to the value of y.
    for t=1:NN
        if (X0(t,1) < y)
            X0(t,1)=y;
        end
    end

  plot(X0(:,1),X0(:,2),'r.','markersize',5);
    axis([0,1,0,1]);
  axis manual;

    if (mod(t,10)==0)
        filename=strcat('pics/',int2str(t),'.png');
        saveas(gcf,filename)
    end
    j=j+1;
   M(j)=getframe(gcf);

end
movie2avi(M,'shrinking.avi');
```