A STUDY OF TYPE-2 FUZZY CLUSTERING

A Thesis

presented to

the Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

WATCHANAN CHANTAPAKUL

Dr. James M. Keller, Thesis Supervisor

MAY 2022

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

A STUDY OF TYPE-2 FUZZY CLUSTERING

presented by Watchanan Chantapakul,
a candidate for the degree of Master of Science in Computer Science,
and hereby certify that, in their opinion, it is worthy of acceptance.

Professor James M. Keller

Professor Derek T. Anderson

Professor Mihail Popescu

DEDICATION

I dedicate my thesis work to my father and my mother who have been my support for all the years of my life. Thank you for your love.

ACKNOWLEDGEMENTS

I would like to give thanks to my thesis supervisor, Dr. James M. Keller, for providing invaluable guidance and much appreciated advice throughout this research. It has been a wonderful opportunity to work with you and learn from you. I am very grateful for your precious time you spent with me.

I would like to thank the committee members, Dr. Derek T. Anderson and Dr. Mihail Popescu, for their academic advice and valuable comments.

I would like to extend my sincere thanks to Dr. Marjorie Skubic and Dr. Rachel M. Proffitt for providing me an opportunity to work as a graduate research assistant with Center for Eldercare and Rehabilitation Technology (CERT), University of Missouri.

I would like to express my sincere gratitude to Dr. Sansanee Auephanwiriyakul for introducing me to the field of fuzzy sets and systems. You have provided several golden opportunities in my life.

I would like to thank Thailand - United States Educational Foundation (TUSEF/Fulbright Thailand) for providing financial support that allowed me to pursue the Master's degree.

Last, but not least, thanks to the excellent academic environment offered at the University of Missouri.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

A STUDY OF TYPE-2 FUZZY CLUSTERING

Watchanan Chantapakul

Dr. James M. Keller, Thesis Supervisor

ABSTRACT

Fuzzy C-means (FCM) has been a prominent clustering algorithm for a long time. It was extended to a type-2 framework by the linguistic fuzzy C-means (LFCM) algorithm that operates on vectors of fuzzy numbers utilizing the extension principle, the decomposition theorem, and interval analyses. The purpose of this thesis is to investigate the iterative type-2 fuzzy clustering algorithms. The LFCM incorporates uncertainty through type-2 fuzzy sets, but it is prone to membership spread, i.e., the uncertainty in a membership function can become too large or broad during the iterative alternating optimization procedure. We devise three dampening approaches to mitigate this problem. Vertical cut dampening, linear dampening, and reflection dampening are defined along with the experiments conducted on a synthetic dataset named the butterfly dataset. We also illustrate the updated memberships (fuzzy numbers) and the resulting cluster prototypes (fuzzy vectors) from visual standpoints. Applying any of these dampening approaches will result in thinner membership functions and helps us control the uncertainty, and in fact, aid in convergence. The linguistic possibilistic C-means (LPCM), which is a type-2 version of the possibilistic C-means (PCM) algorithm, is also studied, and compared to LFCM. We also address some practical guidelines for putting the type-2 fuzzy clustering algorithms into action.

Chapter 1

INTRODUCTION

1.1 Clustering

Pattern recognition (PR) problems are interesting and challenging because they closely relate to human daily life. For instance, people identification from imagery can be useful for security automation. We can categorize PR problems into two main categories: supervised and unsupervised schemes. The former means we have the knowledge about the answers of the problems, and those answers are used as ingredients to teach algorithms how to do recognition. The latter does not make use of what we call label or ground truth that requires a lot of human work, i.e., we specify what the answers should be. Hence, unsupervised algorithms are utilized to discover underlying structure based on some criteria.

Clustering is a big topic under the umbrella of unsupervised learning. It refers to techniques that assemble groups of related data. There are several branches of clustering such as hierarchical clustering and iterative clustering. The procedure of the latter is illustrated in Figure 2.5. The universally recognized iterative clustering algorithm is the Hard C-means (HCM) or the so-called K-means [1] which was devised by MacQueen in 1967. Given a dataset $\{\vec{X}_j\}$, and a set of C cluster centers $\{\vec{C}_i\}$, HCM seeks to find a crisp C-partition that indicates how a dataset can be grouped in C clusters. In the view of similarity, a pattern can only be either not similar (zero) or similar (one) to a cluster center based on the relative (Euclidean) distances. This is defined in terms of the characteristic function $u_{ji}\colon \mathbf{X} \to \{0,1\}$ where $\mathbf{X}$ is the finite universal set. After that, the fuzzy C-means (FCM) algorithm [2] was formalized to raise the granularity of the characteristic function by introducing a concept of fuzzy sets. Instead of having a membership value that can either be zero or one, it can be a value between zero and one. Based on this, the characteristic function is generalized to be the membership function $u_{ji}\colon \mathbf{X} \to [0,1]$. Therefore, a hard C-partition turns into

a fuzzy C-partition with an obligatory constraint that is the sum of memberships of a pattern must be equal to 1, in fact a probabilistic constraint.

There are many endeavors to further enrich the FCM by incorporating uncertainty. This is feasible through the lens of fuzzy sets of type-2. Thus, the FCM which operates under the idea of type-1 fuzzy sets can be extended to type-2 fuzzy C-means (T2-FCM) that, obviously, operates on type-2 fuzzy sets. Rhee and Hwang proposed a type-2 FCM [3] that modifies the FCM (type-1) memberships to have the associated type-2 memberships. The larger memberships have more weights (contributions) than that of the smaller memberships. They also developed an interval type-2 FCM (IT2-FCM) [4] that utilizes two fuzzifiers to generate a footprint of uncertainty. Rubio and Castillo came up with the idea of integrating the FCM with the possibilistic C-means (PCM) [5] which gives rise to another version of IT2-FCM with only one fuzzifier required. It makes use of the memberships and the typicalities to form ranges of memberships. The concept of intuitionistic fuzzy sets were used in [6] to achieve an intuitionistic type-2 FCM, so we get lower bounds and upper bounds of memberships. Auephanwiriyakul and Keller devised the linguistic fuzzy C-means (LFCM) [7]. Unlike those IT2-FCM algorithms, the LFCM provides an alternative way to model uncertainty through fuzzy numbers and fuzzy vectors based on the extension principle, the decomposition theorem, and interval analyses. Cluster centers are linguistic vectors and hence they are also type-2. So, the LFCM can be viewed as a type-2 FCM, or an IT2-FCM if we take only one level cut. However, if we put the LFCM into action, it is susceptible to a problem of membership spread, i.e., a membership function has a long tail.

The linguistic possibilistic C-means (LPCM) [8] is another extension of the regular PCM clustering algorithm. Concerning the membership update for LFCM, u_{ji} depends solely on the squared distance from $\vec{X}_j$ to $\vec{C}_i$. This allows it to handle noise points that might exist in the dataset. The other counterparts of LPCM such as the cluster center update and stopping criterion are similar to that of LFCM.

1.2 Contribution

In this thesis, we are interested in mainly two clustering algorithms named linguistic fuzzy C-means (LFCM), and linguistic possibilistic C-means (LPCM). We study their characteristics, and also how to make them work well in practice. The 15-point butterfly dataset is used to run LFCM with different fuzzifier m values. The cluster centers at different iterations are illustrated. As we examine the LFCM memberships, we can obviously see that the lower level cuts of the memberships become large

very quickly, i.e., the uncertainty increases rapidly, in a few iterations. We call this the membership spread problem. So, the uncertainty associated with the α-cuts are informative at some α values. The dampening approaches are then developed to mitigate the membership spread problem found in the the LFCM clustering algorithm. The first dampening approach is the vertical cut dampening. It is the simplest method to develop and apply to the memberships. By vertically cutting the membership function at a certain level, the uncertainty of the lower level cuts are reduced. The second method is the linear dampening. It works by estimating two slopes, left and right, with respect to the peak of the membership, and then create two lines. This transforms the original membership function into a triangular membership function. We can also multiply some small values to dampen the slopes. The third approach is the reflection dampening. It arises from the fact that we would like to preserve the original shape of the membership. Thus, the reflection dampening takes the side with the lesser uncertainty and reflects it around the peak to the other side. The comparison between the original LFCM and the dampened LFCM is described.

However, LFCM does not handle noise very well. So, LPCM comes into play. The dampening methods can also be applied to the LPCM clustering algorithm in the same manner, even they have different membership update equations.

Based on this research, there is a conference paper, entitled "Controlling Membership Spread in Type-2 Fuzzy Clustering", that has been accepted for publication in the 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2022) at the 2022 IEEE World Congress on Computational Intelligence (IEEE WCCI 2022). Its authors are Watchanan Chantapakul, James M. Keller, and Sansanee Auephanwiriyakul.

Chapter 2

BACKGROUND

2.1 Fuzzy Sets

Computational intelligence (CI) is a field of study that has three main sub-fields under it, including neural networks, fuzzy systems, and evolutionary computation. Fuzzy sets, devised by Zadeh [9], is the core of the fuzzy systems sub-field. Let us review some relevant fuzzy set concepts in this section. We refer the reader to [10] for details of the concepts contained herein.

A fuzzy set A is defined by the membership function

$$\mu_A \colon \mathbf{X} \to [0, 1] \tag{2.1}$$

where $\mathbf{X}$ denotes the universe of discourse such as the real line $\mathbb{R}$. The function $\mu_A(x)$ is typically used interchangeably with $A(x)$.

We define an α-cut of a fuzzy set A as a crisp set of all $x \in \mathbf{X}$ for which the degrees of membership are at least α. It is mathematically defined as

$$^{\alpha}A = \{x \in \mathbf{X} \mid A(x) \geq \alpha\}. \tag{2.2}$$

1-cut of A or ^{1}A has another name called the core of A.

Following the notion of α-cut, the strong α-cut of fuzzy set A is given by

$$^{\alpha+}A = \{x \in \mathbf{X} \mid A(x) > \alpha\}. \tag{2.3}$$

So, the support of fuzzy number A is actually ^{0+}A or the strong 0-cut of A.

A special fuzzy set

$$ {}_{\alpha}A(x) = \alpha \cdot {}^{\alpha}A(x) \tag{2.4} $$

that is closely linked to α-cut is needed to formulate the first decomposition theorem [10]. It states that

$$ A = \bigcup_{\alpha \in [0,1]} {}_{\alpha}A \tag{2.5} $$

where $\bigcup$ is a fuzzy union. This means that we take all level cuts and build fuzzy set based on the decomposition theorem as shown in Figure 2.1.

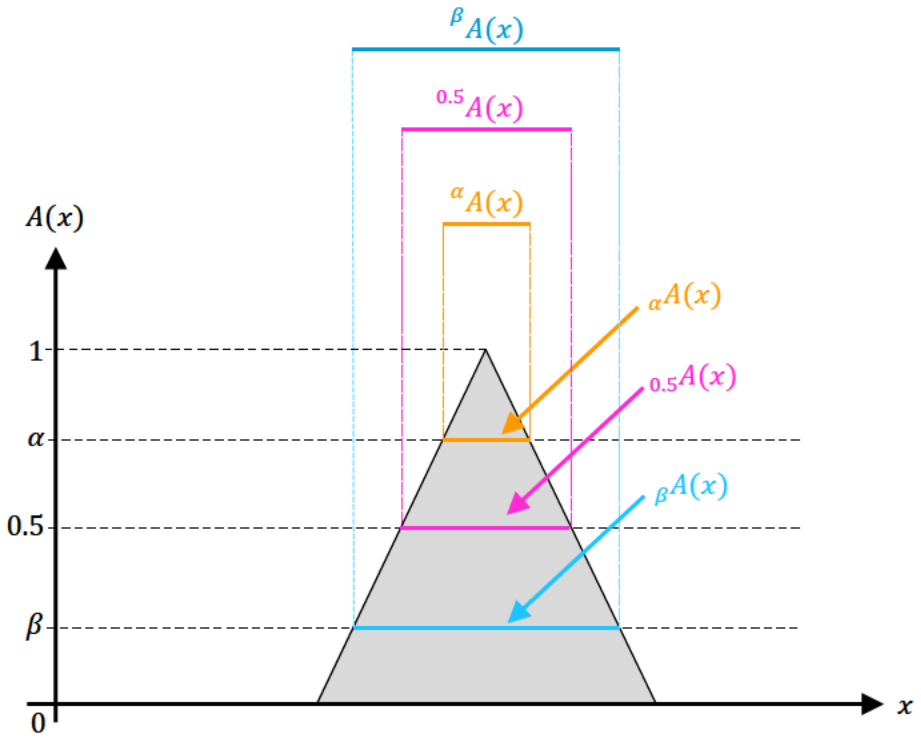


Figure 2.1: The decomposition theorem.

All distinct α values used to represent fuzzy set A is

$$ \Lambda(A) = \{\alpha \mid \exists x \in \mathbf{X}, A(x) = \alpha\} \tag{2.6} $$

which is named as the level set of fuzzy set A. For the experiments in this study, we have $|\Lambda(A)| = 201$, i.e., we use the step of discretization of 0.005 that yields 201 α-cuts for cutting membership grades from 0 to 1. However, one could select the number of level cuts arbitrarily.

Fuzzy numbers used in type-2 fuzzy clustering algorithms must satisfy three properties [10]:

1. A must be a normal fuzzy subset of the real line $\mathbb{R}$.

2. ${}^{\alpha}A$ must be a closed interval, $\forall \alpha \in (0, 1]$.

3. The support of A, ${}^{0+}A$, must be bounded.

Let us define the height of a fuzzy set A to be

$$h(A) = \sup_{x \in \mathbf{X}} A(x). \tag{2.7}$$

Note that we use the supremum (sup) operator here to handle the case of an infinite domain where the maximum may not exist. Following the height of a fuzzy set, A is a normal fuzzy set if $h(A) = 1$.

In addition, there is a special type of fuzzy number named singleton fuzzy number. It serves as a tool to represent a precise regular number in the form of a fuzzy set. We will use it when we are certain about the number for some special occasions. To represent a real number x^* as a singleton fuzzy number A, A needs to satisfy $A(x) = 0, \forall x \in \mathbf{X} \setminus x^*$ and $A(x^*) = 1$. As mentioned before, we will exploit α-cuts to constitute a fuzzy number. So, a singleton fuzzy number can be defined alternatively in terms of level cuts as follows:

$${}^{\alpha}A = [x^*, x^*], \forall \alpha \in [0, 1] \tag{2.8}$$

Figure 2.2 displays a singleton fuzzy number at x^*.

Figure 2.2: Singleton fuzzy number at x^*.

Here, we choose to fuzzify every real number to be in the form of triangular function T. We define a, b, and c to denote the locations of the rising edge, the peak, and the falling edge, respectively. An

α-cut of a triangular fuzzy number can be obtained through:

$$^{\alpha}T(a,b,c) = [\alpha(b-a)+a, c-\alpha(c-b)] \tag{2.9}$$

where $a \leq b \leq c$ and $\alpha \in [0,1]$. The left width of triangular fuzzy number A, $w_l(A)$, is $b-a$. The right width of triangular fuzzy number A, $w_r(A)$, is $c-b$. But from now on, we will use the same value for the left and right widths, $w_l(A) = w_r(A) = w$. Then, we compose a triangular fuzzy number using the decomposition theorem given in (2.5). Figure 2.3 is a visualization of triangular fuzzy number.

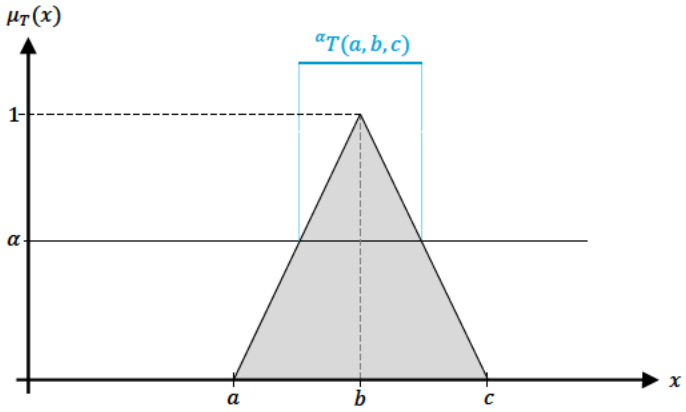


Figure 2.3: Triangular function used to define a fuzzy number.

Let us briefly describe the concept of the extension principle. It is a fundamental tool to transform any crisp function into a fuzzy function. Since we are concerned about arithmetic operations, we focus on binary operators. A binary operator is a function $f\colon \mathbf{X}_1 \times \mathbf{X}_2 \to \mathbf{Y}$ that takes two arguments, x_1 and x_2 in the universes of discourse $\mathbf{X}_1$ and $\mathbf{X}_2$, respectively, where $\times$ denotes the Cartesian product. Using the extension principle, we it turns into a fuzzy binary operator that operates on two fuzzy numbers (A_1 and A_1) and yields a fuzzy number B. We define it as:

$$f(A_1, A_2)(y) = \sup\{A_1(x_1) \wedge A_2(x_2) \mid y = f(x_1, x_2), x_1 \in \mathbf{X}_1, x_2 \in \mathbf{X}_2\} \tag{2.10}$$

In general, the extension principle works with any number of domains and any fuzzy sets (they do not need to be fuzzy numbers). But we show only the specific case for fuzzy arithmetic.

Note that, in practice, the higher levels of discretization, the more it becomes computationally intractable to directly compute with the extension principle. The concept of α-cut is predominant in representing any fuzzy set by discretizing its membership grades into crisp sets as opposed to

discretizing the universe of discourse $\mathbf{X}$. Therefore, instead of directly computing the extended crisp functions using the extension principle, we use the extension principle to extend arithmetic operations on real numbers to fuzzy numbers.

To implement LFCM or LPCM, it involves arithmetic operations between two fuzzy numbers, $A\#B$ where $\# \in \{+, -, \cdot, /\}$, at a given α-cut, is determined as [10]

$$ {}^{\alpha}[A\#B] = [{}^{\alpha}A] \# [{}^{\alpha}B] . \tag{2.11} $$

In other words, it can be computed using interval arithmetic rather than fuzzy arithmetic because the α-cuts are just closed intervals.

Now, combining (2.4), (2.5) and (2.11) together allows us to perform fuzzy arithmetic on two fuzzy numbers. So, we obtain

$$ \begin{aligned} A\#B &= \bigcup_{\alpha\in[0,1]} {}_{\alpha}[A\#B] \\ &= \bigcup_{\alpha\in[0,1]} ([_{\alpha}A] \# [_{\alpha}B]) . \end{aligned} \tag{2.12} $$

The type-2 fuzzy clustering algorithms investigated in this thesis are LFCM and LPCM. Now, we define a fuzzy vector as a p-dimensional vector of fuzzy numbers, or $\vec{A} = \langle A_1, A_2, \ldots, A_p \rangle$, as illustrated in Figure 2.4. So, its elements are fuzzy numbers A_k associated with the indexes $k = 1, \ldots, p$. A fuzzy vector is also called a noninteractive linguistic vector. The properties of fuzzy vectors are described in [7]. A fuzzy vector is actually a type-2 fuzzy set over the index set $K = \{1, 2, \ldots, p\}$ because there is a type-1 fuzzy set A_k associated with index k. Therefore, the LFCM (LPCM) can be considered as a type-2 version of the FCM (PCM).

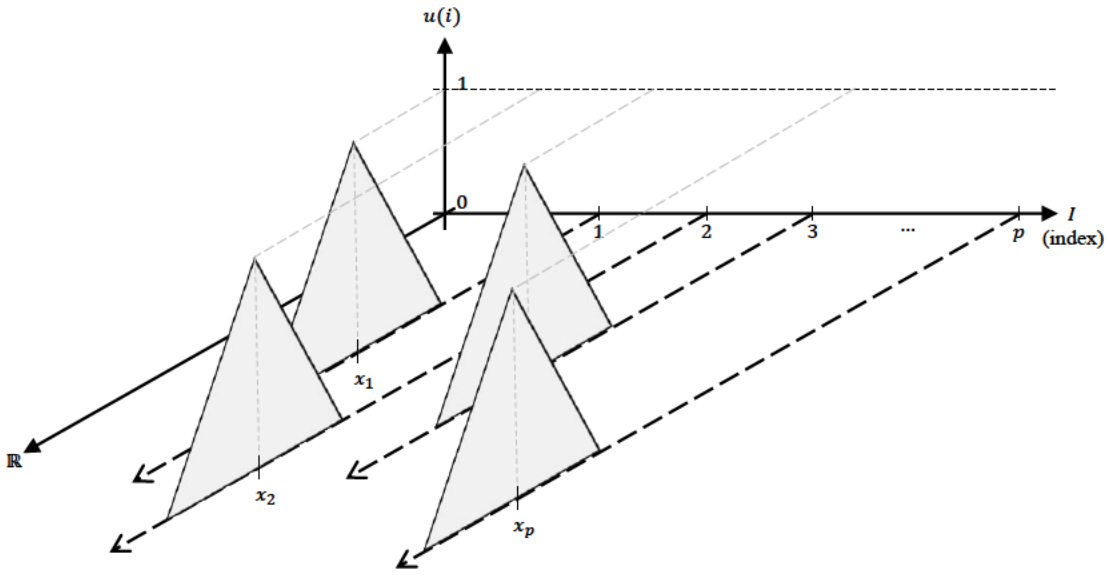


Figure 2.4: Visualization of fuzzy vector.

Since we will deal with the uncertainty associated with fuzzy numbers and fuzzy vectors in clustering, now discuss a way to measure the amount of uncertainty of a fuzzy set A. The nonspecificity of non-empty fuzzy set A can be measured using the generalized Hartley function [10]

$$U(A) = \frac{1}{h(A)} \int_0^{h(A)} \log\left(1 + m\left({}^{\alpha}A\right)\right) d\alpha \tag{2.13}$$

where $m\left({}^{\alpha}A\right)$ denotes a measure of the α-cut of A. We transform the above equation into a discrete form as follows:

$$U(A) = \frac{1}{h(A)} \sum_{i=1}^{|\Lambda(A)|} \log\left(1 + m\left({}^{\alpha}A\right)\right)(\alpha_i - \alpha_{i+1}) \tag{2.14}$$

where $\alpha_i > \alpha_{i+1}$ and $\alpha_{|\Lambda(A)|+1} = 0$. We call (2.14) the U-uncertainty of non-empty fuzzy set A. Singleton fuzzy number and fuzzy number defined in this study are, of course, non-empty fuzzy set.

Recall that the α-cut of A is always a closed interval ${}^{\alpha}A = [a, b]$. According to measure theory, length of the interval is $b - a$. Hence, we simply replace the measure of the α-cut of A in (2.14) with

$$m\left({}^{\alpha}A\right) = b - a. \tag{2.15}$$

Also, note that we use the natural logarithm here rather than the binary log. Since $m\left({}^{\alpha}A\right) \in \mathbb{R}_{\geq 0}$ and $\log(0)$ is undefined, we add one inside the log to prevent this error. For a singleton fuzzy number, the interval widths for all α-cuts are zero (and $\log(1 + 0) = 0$). Thus, the U-uncertainty of singleton fuzzy number is equal to zero. That is, it has the lowest possible uncertainty. One can use $\log_2$

instead of log to measure the uncertainty of each interval in bits.

Now, the U-uncertainty is ready to be used as a tool for measuring uncertainty of fuzzy numbers. Also, we need to measure the amount of uncertainty of a p-dimensional fuzzy vector $\vec{A}$. Therefore, we can define the U-uncertainty of $\vec{A}$ to be

$$U(\vec{A}) = \frac{1}{p}\sum_{i=1}^{p} U(A_i) \tag{2.16}$$

or

$$U(\vec{A}) = \max_{i=1,\ldots,p} U(A_i). \tag{2.17}$$

where A_i is a fuzzy number in dimension i. They make more sense than using just the sum of U-uncertainties of fuzzy numbers. The reason is that they allow us to compare the amounts of uncertainty of fuzzy vectors that have different number of dimensions. The former represents the average uncertainty of fuzzy numbers in the fuzzy vector. The latter can be useful when at least one fuzzy number of the fuzzy vector is uncertain so the fuzzy vector is also considered uncertain. In this study, we choose (2.16) to measure the uncertainty of a fuzzy vector.

2.2 Interval Analyses

As described above, the decomposition theorem is used to build fuzzy numbers. For any fuzzy numbers, α-cuts of them for $\alpha \in (0, 1]$ are always closed intervals, Thus, a closed interval, I, can be defined by

$$I = [a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\} \tag{2.18}$$

where a denotes the lower endpoint and b represents the upper endpoint of the closed interval I.

There are some interval functions needed to execute the LFCM and the LPCM. The width of closed interval I is denoted by

$$\text{width}(I) = b - a. \tag{2.19}$$

The middle point of closed interval I is defined as

$$\text{mid}(I) = \frac{b + a}{2}. \tag{2.20}$$

The magnitude of an interval I which is the maximum absolute value in an interval is mathematically

given by

$$\text{mag}(I) = \max\{|x| \mid x \in I\}. \tag{2.21}$$

So, note that $|\cdot|$ denotes the symbol of absolute value.

Since the LFCM involves calculating fuzzy Euclidean distances, we need to define the following interval functions: the square root function and the square function. The square root of the closed interval i is defined by

$$\sqrt{I} = I^{1/2} = \left[\sqrt{a}, \sqrt{b}\right] \text{ for } a \geq 0 \tag{2.22}$$

The square function for a closed interval I can be obtained by

$$I^2 = \begin{cases} \left[0, \text{mag}(I)^2\right] & \text{if } 0 \in X \\ \left[a^2, b^2\right] & \text{if } a^2 \leq b^2 \\ \left[b^2, a^2\right] & \text{if } a^2 \geq b^2 \end{cases}. \tag{2.23}$$

We can see that the square function defined in Equation 2.23 requires the magnitude function given by Equation 2.21. Therefore, we can further simplify the first condition of Equation 2.23 [11]. If the input to the magnitude function is the square function, $f(a) = a^2$ where $a \in \mathbb{R}$, the magnitude function then becomes

$$\text{mag}(f(I)) = \max\{(a)^2, (b)^2\} \tag{2.24}$$

as we make use of the extrema. Thus, we modify Equation 2.23 to be as follows:

$$I^2 = \begin{cases} \left[0, \max(a^2, b^2)\right] & \text{if } 0 \in X \\ \left[a^2, b^2\right] & \text{if } a^2 \leq b^2 \\ \left[b^2, a^2\right] & \text{if } a^2 \geq b^2 \end{cases} \tag{2.25}$$

Apart from that, interval arithmetic is also needed [7]. Let us define two intervals $I_1 = [a, b]$ and $I_2 = [c, d]$. Let r be a real number. The following are some important definitions of interval arithmetic:

$$r \cdot I_1 = r \cdot [a, b] = \begin{cases} [ra, rb] & \text{if } r > 0 \\ [rb, ra] & \text{if } r < 0 \end{cases}, \tag{2.26}$$

$$I_1 + I_2 = [a, b] + [c, d] = [a + c, b + d], \tag{2.27}$$

$$I_1 - I_2 = [a, b] - [c, d] = [a - d, b - c], \tag{2.28}$$

$$I_1 \cdot I_2 = [a, b] \cdot [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)], \tag{2.29}$$

$$I_1 / I_2 = [a, b]/[c, d] = [a, b] \times \left[\frac{1}{d}, \frac{1}{c}\right], 0 \notin I_2. \tag{2.30}$$

See also [11] and [8] for more details of interval analyses.

2.3 Linguistic Fuzzy C-Means

K-means is one of the most popular iterative clustering algorithms that was proposed by MacQueen [1] in 1967. Figure 2.5 shows the standard procedure for an iterative clustering algorithm.

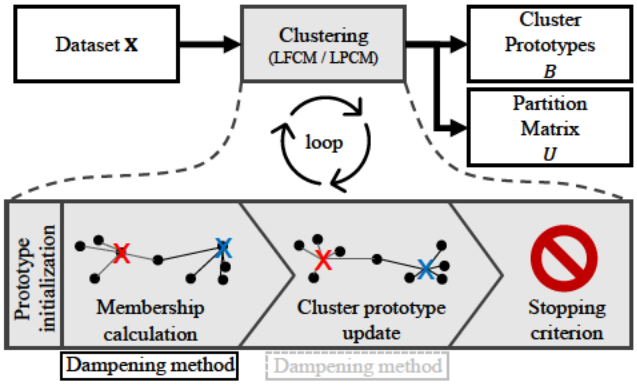


Figure 2.5: Iterative clustering scheme.

First, we process N patterns in the dataset X by repeating three steps multiple times. The steps includes 1) membership calculation, 2) cluster prototype update, and 3) stopping criterion. The outputs of iterative clustering are cluster prototypes B and partition matrix U. Cluster prototypes or cluster centers are the representatives of different clusters. Clustering can also be used to perform classification, for example, given cluster prototypes, we compute the distance between a new pattern with the existing cluster prototypes, it can be classified as the same class or group as the closest prototype. The idea of closeness here plays a role of similarity. If a pair of patterns has a small distance, meaning they are close to each other, they are similar.

The fuzzy C-means (FCM) [2] is a widely recognized clustering algorithm as a generalized version of the HCM. To incorporate the concept of uncertainty to the FCM, the linguistic fuzzy C-means

(LFCM), developed by Auephanwiriyakul and Keller [7], as a version of type-2 fuzzy C-means enables us to do so. Let $X = \{\vec{x}_j\}$ denote the dataset consisting of N patterns where each is a p-dimensional vector, $\vec{x}_k \in \mathbb{R}^p$, and a C-tuple of cluster centers $B = \langle \vec{c}_i \rangle$ where $\vec{c}_i \in \mathbb{R}^p$. Let $U = [u_{ji}]_{C \times N}$ be a partition matrix accommodating the membership grade $u_{ji} \in [0, 1]$ that the pattern $\vec{x}_j$ belongs to the cluster A_i. The regular FCM involves minimizing the objective function which is

$$J(B, U; X) = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ji})^m (d_{ji})^2 \tag{2.31}$$

where $(d_{ji})^2 = d^2(\vec{x}_j, \vec{c}_i)$ is the squared distance between pattern $\vec{x}_j$ and cluster center $\vec{c}_i$, and for some fuzzifier $m > 1$ that controls how fuzzy the system is through the partition matrix. This is a minimization of weighted sum of fuzzy squared distances between the patterns and the cluster centers. The criterion function (2.31) is subject to the constraint

$$\sum_{i=1}^{C} u_{ji} = 1, \forall j = 1, \ldots, N. \tag{2.32}$$

It compels each pattern to have grades of membership to all clusters, and the sum of them is unity. Note that setting $m = 1$ is invalid and does not give us the HCM. However, the limit as m approaches 1 from above does converge to the HCM. The necessary condition for updating the FCM membership u_{ji} (the degree of membership of $\vec{x}_j$ in the i^{th} cluster), is

$$\left.\begin{array}{ll} u_{ji} = \frac{1}{\sum_{k=1}^{C} \left(\frac{(d_{ji})^2}{(d_{jk})^2} \right)^{1/(m-1)}}, & \text{if } Z_j = \emptyset \\ \left.\begin{array}{ll} u_{ji} = 0 & \text{if } i \notin Z_j \\ \sum_{i \in Z_j}^{u_{ji}} = 1 & \text{if } i \in Z_j \end{array}\right\}, & \text{if } Z_j \neq \emptyset \end{array}\right\} \tag{2.33}$$

where $Z_j = \{i \mid 1 \leq i \leq C, (d_{ji})^2 = 0\}$.

The cluster center update equation of $\vec{C}_i$ is given by

$$\vec{c}_i = \frac{\sum_{j=1}^{N} (u_{ji})^m \vec{x}_j}{\sum_{j=1}^{N} (u_{ji})^m}. \tag{2.34}$$

When it comes to the LFCM, we perform the alternating optimization by modifying the necessary conditions of the real number version in (2.33) and (2.34). It is technically done by substituting real numbers with fuzzy numbers, and real vectors with fuzzy vectors. Consequently, u_{ji} and d_{ji} also become fuzzy numbers, whereas $\vec{x}_j$ and $\vec{c}_i$ turn into fuzzy vectors $\vec{X}_j$ and $\vec{C}_i$, respectively. Hence,

$\mathbf{X} = \{\vec{X}_j\}$ is the dataset consisting of N fuzzy vectors. Each input pattern $\vec{X}_j = \langle X_{j1}, X_{j2}, \ldots, X_{jp}\rangle \in \mathcal{P}(\mathbb{R})^p$ is a vector of p fuzzy numbers, where $\mathcal{P}(\mathbb{R})$ denotes the fuzzy power set of $\mathbb{R}$. Also, $B = \langle \vec{C}_i\rangle$ is a C-tuple of cluster centers where the cluster center $\vec{C}_i = \langle C_{i1}, C_{i2}, \ldots, C_{ip}\rangle \in \mathcal{P}(\mathbb{R})^p$ is a vector of p fuzzy numbers. Now, each element in the partition matrix $U = [u_{ji}]_{N\times C}$ is non-negative fuzzy number instead of non-negative real number.

The LFCM algorithm is summarized in Algorithm 1 (note that this algorithm is also the algorithm for the classical FCM.). We can see that there are five key steps: 1) cluster prototype initialization, 2) computing fuzzy Euclidean distance, 3) membership calculation, 4) cluster prototype update, and 5) termination. We will go over each of the steps, except the first one since the cluster prototypes will be simply manually selected in this work. Refer to [8] for more information regarding the proofs of LFCM equations.

Algorithm 1: Linguistic fuzzy C-means (LFCM)

Input: The number of clusters C, the dataset $\mathbf{X}$
Output: The C-tuple of prototypes B, the C-partition matrix U

Initialize cluster prototypes $\vec{C}_i$, $i = 1, \ldots, C$
repeat
  Compute fuzzy Euclidean distance $d_{ji}, \forall j, i$ using (2.37) and (2.39)
  Compute membership $u_{ji}, \forall j, i$ using (2.41) and (2.43)
  Update cluster prototype $\vec{C}_i, \forall i$ using Algorithm 2
until *Convergence*;

2.3.1 Fuzzy Euclidean Distance

Calculating the memberships of the LFCM are based on the relative distances between patterns and cluster centers. Hence, we need the definition of fuzzy Euclidean distance between two fuzzy vectors. Recall that the definition of the Euclidean distance between two p-dimensional vectors, $\vec{x}, \vec{y} \in \mathbb{R}^p$ is given by

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{k=1}^{p}(x_k - y_k)^2} \tag{2.35}$$

where x_k is the k-th element in $\vec{x}$, and y_k is the k-th element in $\vec{y}$, for all $k = 1, \ldots, p$. In the framework of LFCM, we need to modify (2.35) to work with fuzzy numbers and fuzzy vectors. Thus, the extension principle is our tool here. Let us define two p-dimensional fuzzy vectors, $\vec{A} = \langle A_1, \ldots, A_p\rangle$ and $\vec{B} = \langle B_1, \ldots, B_p\rangle$. (2.35) is extended to be

$$d(\vec{A}, \vec{B}) = \sqrt{\sum_{k=1}^{p}(A_k - B_k)^2}. \tag{2.36}$$

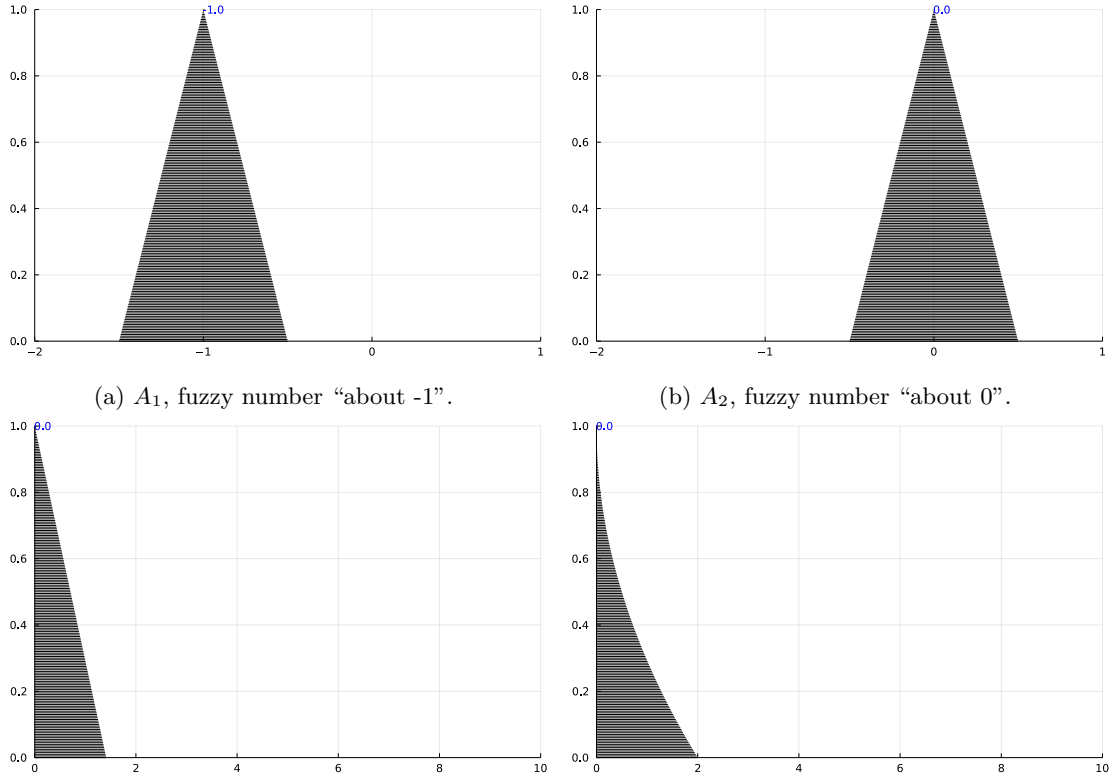
As we make use of the decomposition theorem to build fuzzy number from α-cuts, the fuzzy Euclidean distance is defined as

$$d(\vec{A}, \vec{B}) = \begin{cases} \mathbf{0} & \text{if } \vec{A} = \vec{B} \\ \bigcup_{\alpha \in [0,1]} {}^{\alpha}[d(\vec{A}, \vec{B})] & \text{otherwise} \end{cases}, \tag{2.37}$$

where $\mathbf{0}$ is singleton fuzzy number zero, and each level cut of $d(\vec{A}, \vec{B})$, is

$${}^{\alpha}[d(\vec{A}, \vec{B})] = \left[\sum_{k=1}^{p} ({}^{\alpha}A_k - {}^{\alpha}B_k)^2 \right]^{1/2}. \tag{2.38}$$

The first condition in (2.37) is needed to handle the case where $\vec{A}$ and $\vec{B}$ are identical. The reason is that if we directly compute the the fuzzy distance using interval arithmetic, we will get fuzzy number "about 0" as displayed in Figure 2.6, not singleton fuzzy number 0. Hence, it requires the special condition.



(a) A_1, fuzzy number "about -1". (b) A_2, fuzzy number "about 0".

(c) $d(\vec{A}, \vec{A})$. (d) $(d(\vec{A}, \vec{A}))^2$.

Figure 2.6: Fuzzy Euclidean distance $d(\vec{A}, \vec{A})$ when $\vec{A}$ is fuzzy vector ⟨"about -1", "about 0"⟩.

In LFCM, the fuzzy Euclidean distance is used to calculate memberships u_{ji} that is the degree of belonging of fuzzy vector $\vec{X}_j$ in the i^{th} cluster. We can then change (2.38) to compute the fuzzy

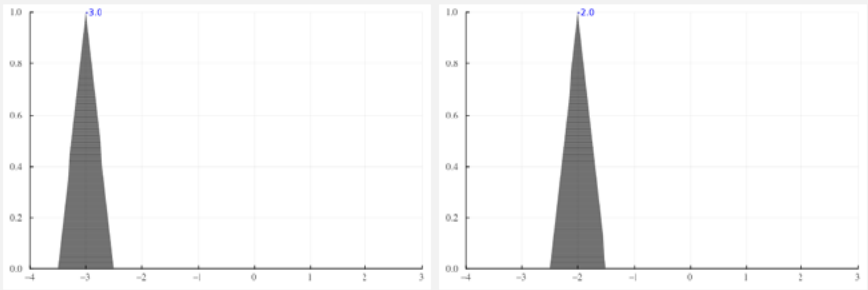
distance between $\vec{X}_j$ and $\vec{C}_i$ by the formula

$$
\begin{aligned}
{}^{\alpha}[d_{ji}] = {}^{\alpha}\left[d(\vec{X}_j, \vec{C}_i)\right] &= \left[({}^{\alpha}X_{j1} - {}^{\alpha}C_{i1})^2 + \cdots + ({}^{\alpha}X_{jp} - {}^{\alpha}C_{ip})^2\right]^{1/2} \\
&= \left[\sum_{k=1}^{p} \left[x_{k[l]} - c_{k[r]}, x_{k[r]} - c_{k[l]}\right]^2\right]^{1/2}
\end{aligned} \tag{2.39}
$$

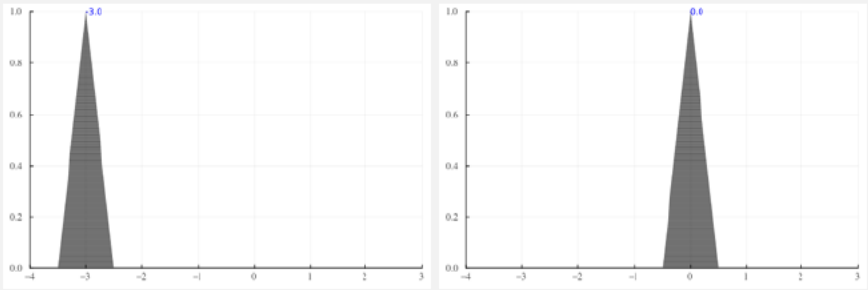
where ${}^{\alpha}X_{jk} = [x_{jk[l]}, x_{jk[r]}]$ is the α-cut of the k^{th} fuzzy number of $\vec{X}_j$. and ${}^{\alpha}C_{ik} = [c_{ik[l]}, c_{ik[r]}]$ is the α-cut of the k^{th} fuzzy number of $\vec{C}_i$. To compute (2.39), we use (2.25), (2.22) and (2.28) which are discussed earlier.

Example 2.1: Computing fuzzy Euclidean distance.

Now consider a numerical example of computing fuzzy Euclidean distance between two fuzzy vectors $\vec{A} = \langle$"about -3", "about -2"$\rangle$ and $\vec{A} = \langle$"about -3", "about 0"$\rangle$ shown in Figure 2.7.



(a) A_1, fuzzy number "about -3". (b) A_2, fuzzy number "about -2".



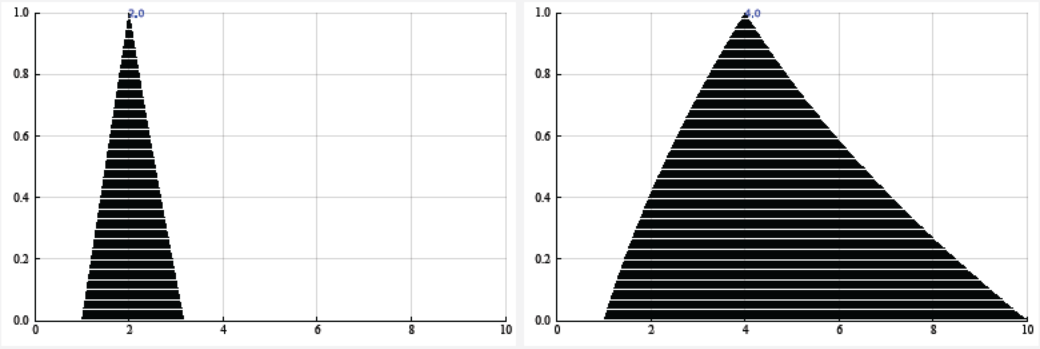
(c) B_1, fuzzy number "about -3". (d) B_1, fuzzy number "about 0".

Figure 2.7: Fuzzy vectors $\vec{A} = \langle A_1, A_2\rangle$ and $\vec{B} = \langle B_1, B_2\rangle$.

For $\alpha = 0$, ${}^{\alpha}\vec{A} = [[-3.5, -2.5], [-2.5, -1.5]]$, ${}^{\alpha}\vec{B} = [[-3.5, -2.5], [-0.5, 0.5]]$.

$$
\begin{aligned}
{}^{\alpha}[d(\vec{A}, \vec{B})] &= \left[({}^{\alpha}A_1 - {}^{\alpha}B_1)^2 + ({}^{\alpha}A_2 - {}^{\alpha}B_2)^2\right]^{1/2} \\
&= \left[([-3.5, -2.5] - [-3.5, -2.5])^2 + ([-2.5, -1.5] - [-0.5, 0.5])^2\right]^{1/2} \\
&= \left[[-3.5 - (-2.5), -2.5 - (-3.5)]^2 + [-2.5 - 0.5, -1.5 - (-0.5)]^2\right]^{1/2} \\
&= \left[[-1, 1]^2 + [-3, -1]^2\right]^{1/2} \\
&= [[0, 1] + [1, 9]]^{1/2} \\
&= [1, 10]^{1/2} \\
&= [\sqrt{1}, \sqrt{10}] \\
&= [1.00, 3.16]
\end{aligned}
$$

Using the decomposition theorem, the fuzzy distance, $d(\vec{A}, \vec{B})$, and the squared fuzzy distance, $(d(\vec{A}, \vec{B}))^2$, are shown in Figure 2.8.



(a) $d(\vec{A}, \vec{B})$. (b) $(d(\vec{A}, \vec{B}))^2$.

Figure 2.8: Visualization of the fuzzy distance and the squared fuzzy distance.

2.3.2 LFCM Memberships

To extend the FCM membership update equation in (2.33) to the LFCM membership update equation, we will make use of the extension principle. So, we arrive at:

$$
u_{ji} = \left\{ \begin{array}{ll} \dfrac{1}{\sum_{k=1}^{C} \left(\frac{(d_{ji})^2}{(d_{jk})^2} \right)^{1/(m-1)}}, & \text{if } Z_j = \emptyset \\ \left. \begin{array}{ll} \mathbf{0} & \text{if } i \notin Z_j \\ \mathbf{S} & \text{if } i \in Z_j \end{array} \right\}, & \text{if } Z_j \neq \emptyset \end{array} \right. \tag{2.40}
$$

where $(d_{ji})^2$ now is the squared fuzzy Euclidean distance between $\vec{X}_j$ and $\vec{C}_i$, $Z_j = \{i \mid 1 \leq i \leq C \text{ and } (d_{ji})^2 = \mathbf{0}\}$, and $\mathbf{0}$ and $\mathbf{S}$ are the singleton fuzzy numbers at 0 and $1/|Z_j|$, respectively. In other words, Z_j is a set that comprises the indices of patterns that are identical to cluster center $\vec{C}_j$. They are identical if the squared distance is equal to zero in the FCM. However, in the LFCM, we need to check if all level cuts are identical as in the case of fuzzy distance.

We apply the decomposition theorem on (2.40) and arrive at

$$u_{ji} = \begin{cases} \bigcup_{\alpha \in [0,1]} {}^{\alpha}u_{ji}, & \text{if } Z_j = \emptyset \\ \left.\begin{array}{ll} \mathbf{0} & \text{if } i \notin Z_j \\ \mathbf{S} & \text{if } i \in Z_j \end{array}\right\}, & \text{if } Z_j \neq \emptyset \end{cases} \tag{2.41}$$

Therefore we can compute the membership function u_{ji} at different α-cuts and combine them together with some special conditions. Following the equation above, at a given level cut, we use

$${}^{\alpha}u_{ji} = \frac{\left({}^{\alpha}\left[d_{ji}^2\right]\right)^{1/(1-m)}}{\sum_{k=1}^{C}\left({}^{\alpha}\left[d_{jk}^2\right]\right)^{1/(1-m)}} \tag{2.42}$$

to compute α-cuts of u_{ji}. An α-cut of membership u_{ji} yields a closed interval ${}^{\alpha}[u_{ji}] = \left[{}^{\alpha}\left[u_{ji}\right]_{[l]}, {}^{\alpha}\left[u_{ji}\right]_{[r]}\right]$. However, we can make use of the property of continuous and nonincreasing function as proven by [7]. Therefore, given ${}^{\alpha}[d_{ji}^2] = \left[{}^{\alpha}\left[d_{ji}^2\right]_{[l]}, {}^{\alpha}\left[d_{ji}^2\right]_{[r]}\right]$, we can compute the lower and upper endpoints with

$${}^{\alpha}\left[u_{ji}\right]_{[l]} = \frac{\left({}^{\alpha}\left[d_{ji}^2\right]_{[r]}\right)^{h}}{\left({}^{\alpha}\left[d_{ji}^2\right]_{[r]}\right)^{h} + \sum_{k\neq i}^{C}\left({}^{\alpha}\left[d_{jk}^2\right]_{[l]}\right)^{h}}, \tag{2.43a}$$

$${}^{\alpha}\left[u_{ji}\right]_{[r]} = \frac{\left({}^{\alpha}\left[d_{ji}^2\right]_{[l]}\right)^{h}}{\left({}^{\alpha}\left[d_{ji}^2\right]_{[l]}\right)^{h} + \sum_{k\neq i}^{C}\left({}^{\alpha}\left[d_{jk}^2\right]_{[r]}\right)^{h}}, \tag{2.43b}$$

respectively, where $h = 1/(1-m)$. We also need the four corollaries established under Theorem 4.6 in [7] to handle the boundary cases. The special conditions are as follows:

1. If ${}^{\alpha}\left[d_{ji}\right]_{[r]} \neq 0$ and ${}^{\alpha}\left[d_{jk}\right]_{[l]} = 0$, $\exists k \neq i$, then ${}^{\alpha}\left[u_{ji}\right]_{[l]} = 0$.
2. If ${}^{\alpha}\left[d_{ji}\right]_{[l]} = {}^{\alpha}\left[d_{ji}\right]_{[r]} = 0$ and ${}^{\alpha}\left[d_{jk}\right]_{[l]} \neq 0$, $\exists k \neq i$, then ${}^{\alpha}\left[u_{ji}\right]_{[l]} = 1$.
3. If ${}^{\alpha}\left[d_{ji}\right]_{[l]} \neq 0$ and ${}^{\alpha}\left[d_{jk}\right]_{[r]} = 0$, $\exists k \neq i$, then ${}^{\alpha}\left[u_{ji}\right]_{[r]} = 0$.
4. If ${}^{\alpha}\left[d_{ji}\right]_{[l]} = 0$ and ${}^{\alpha}\left[d_{jk}\right]_{[r]} \neq 0$, $\exists k \neq i$, then ${}^{\alpha}\left[u_{ji}\right]_{[r]} = 1$.

Example 2.2: Use of the first and forth special conditions.

Given $\alpha = 0$, $m = 2$, ${}^{\alpha}\left[(d_{5,1})^2\right] = [0, 136]$ and ${}^{\alpha}\left[(d_{5,2})^2\right] = [484, 1205]$.

$$\begin{aligned}
\text{The membership } {}^{\alpha}[u_{5,1}] &= \left[\frac{({}^{\alpha}\left[d^2_{5,1}\right]_{[r]})^{1/(m-1)}}{({}^{\alpha}\left[d^2_{5,1}\right]_{[r]})^{1/(m-1)} + ({}^{\alpha}\left[d^2_{5,2}\right]_{[l]})^{1/(m-1)}}, \right. \\
&\qquad \left. \frac{({}^{\alpha}\left[d^2_{5,1}\right]_{[l]})^{1/(m-1)}}{({}^{\alpha}\left[d^2_{5,1}\right]_{[l]})^{1/(m-1)} + ({}^{\alpha}\left[d^2_{5,2}\right]_{[r]})^{1/(m-1)}}\right] \\
&= \left[\frac{136^{-1}}{136^{-1} + 484^{-1}}, \frac{0^{-1}}{0^{-1} + 1205^{-1}},\right] \longleftarrow \text{invalid} \\
&= [0.78, 1.00] \longleftarrow \text{Corollary 4} \\
\text{The membership } {}^{\alpha}[u_{5,2}] &= \left[\frac{({}^{\alpha}\left[d^2_{5,2}\right]_{[r]})^{1/(m-1)}}{({}^{\alpha}\left[d^2_{5,2}\right]_{[r]})^{1/(m-1)} + ({}^{\alpha}\left[d^2_{5,1}\right]_{[l]})^{1/(m-1)}}, \right. \\
&\qquad \left. \frac{({}^{\alpha}\left[d^2_{5,2}\right]_{[l]})^{1/(m-1)}}{({}^{\alpha}\left[d^2_{5,2}\right]_{[l]})^{1/(m-1)} + ({}^{\alpha}\left[d^2_{5,1}\right]_{[r]})^{1/(m-1)}}\right] \\
&= \left[\frac{1205^{-1}}{1205^{-1} + 0^{-1}}, \frac{484^{-1}}{484^{-1} + 136^{-1}},\right] \longleftarrow \text{invalid} \\
&= [0.00, 0.22] \longleftarrow \text{Corollary 1}
\end{aligned}$$

Example 2.3: Computing LPCM memberships (no corollaries needed).

Given $\alpha = 0$, $m = 2$, ${}^{\alpha}\left[(d_{5,1})^2\right] = [10, 40]$ and ${}^{\alpha}\left[(d_{5,2})^2\right] = [60, 100]$.

$$\begin{aligned}
{}^{\alpha}[u_{5,1}] &= \left[\frac{({}^{\alpha}\left[d^2_{5,1}\right]_{[r]})^{1/(m-1)}}{({}^{\alpha}\left[d^2_{5,1}\right]_{[r]})^{1/(m-1)} + ({}^{\alpha}\left[d^2_{5,2}\right]_{[l]})^{1/(m-1)}}, \frac{({}^{\alpha}\left[d^2_{5,1}\right]_{[l]})^{1/(m-1)}}{({}^{\alpha}\left[d^2_{5,1}\right]_{[l]})^{1/(m-1)} + ({}^{\alpha}\left[d^2_{5,2}\right]_{[r]})^{1/(m-1)}}\right] \\
&= \left[\frac{40^{-1}}{40^{-1} + 60^{-1}}, \frac{10^{-1}}{10^{-1} + 100^{-1}},\right] \\
&= [0.60, 0.91] \\
{}^{\alpha}[u_{5,2}] &= \left[\frac{({}^{\alpha}\left[d^2_{5,2}\right]_{[r]})^{1/(m-1)}}{({}^{\alpha}\left[d^2_{5,2}\right]_{[r]})^{1/(m-1)} + ({}^{\alpha}\left[d^2_{5,1}\right]_{[l]})^{1/(m-1)}}, \frac{({}^{\alpha}\left[d^2_{5,2}\right]_{[l]})^{1/(m-1)}}{({}^{\alpha}\left[d^2_{5,2}\right]_{[l]})^{1/(m-1)} + ({}^{\alpha}\left[d^2_{5,1}\right]_{[r]})^{1/(m-1)}}\right] \\
&= \left[\frac{100^{-1}}{100^{-1} + 10^{-1}}, \frac{60^{-1}}{60^{-1} + 40^{-1}},\right] \\
&= [0.09, 0.40]
\end{aligned}$$

According to the α-cuts of the squared distances given in this example, $\vec{X}_5$ is closer to $\vec{C}_1$ than $\vec{C}_2$. So, $u_{5,1}$ is larger than $u_{5,2}$.

Example 2.4: Uncertainty in LPCM memberships.

Let $\alpha = 0$ and $m = 2$.

1. ${}^{\alpha}\left[(d_{5,1})^2\right] = [10, 40]$ and ${}^{\alpha}\left[(d_{5,2})^2\right] = [10, 40]$

$$\begin{aligned}\text{The membership } {}^{\alpha}[u_{5,1}] &= \left[\frac{40^{-1}}{40^{-1}+10^{-1}}, \frac{10^{-1}}{10^{-1}+40^{-1}}\right] \\ &= [0.20, 0.80] \\ \text{The membership } {}^{\alpha}[u_{5,2}] &= \left[\frac{40^{-1}}{40^{-1}+10^{-1}}, \frac{10^{-1}}{10^{-1}+40^{-1}}\right] \\ &= [0.20, 0.80]\end{aligned}$$

   The uncertainty of the resulting memberships for this level cut is determined by $\log(1 + (0.80 - 0.20)) = 0.47$ nats which is a component in the U-uncertainty.

2. ${}^{\alpha}\left[(d_{5,1})^2\right] = [10, 40]$ and ${}^{\alpha}\left[(d_{5,2})^2\right] = [10, 50]$

$$\begin{aligned}\text{The membership } {}^{\alpha}[u_{5,1}] &= \left[\frac{40^{-1}}{40^{-1}+10^{-1}}, \frac{10^{-1}}{10^{-1}+50^{-1}}\right] \\ &= [0.20, 0.83] \\ \text{The membership } {}^{\alpha}[u_{5,2}] &= \left[\frac{50^{-1}}{50^{-1}+10^{-1}}, \frac{10^{-1}}{10^{-1}+40^{-1}}\right] \\ &= [0.17, 0.80]\end{aligned}$$

   The uncertainty of the resulting memberships for this level cut is determined by $\log(1 + (0.83 - 0.20)) = \log(1 + (0.80 - 0.17)) = 0.49$ nats which is a component in the U-uncertainty.

We can see that ${}^{\alpha}\left[(d_{5,2})^2\right]$ in the first case is more certain. Hence, the resulting memberships contain less uncertainty (0.47 nats) than that of the second case (0.49 nats).

2.3.3 LFCM Prototype Update

To extend the membership equation, the extension principle is applied on (2.34), so we get

$$\vec{C}_i = \frac{\sum_{j=1}^{N} (u_{ji})^m \vec{X}_j}{\sum_{j=1}^{N} (u_{ji})^m} \tag{2.44}$$

Each dimension of $\vec{C}_i = \langle C_{i1}, \ldots, C_{ip} \rangle$ is computed independently. Based on the decomposition theorem, the i^{th} cluster center in the d^{th} dimension, $\vec{C}_{kd}$, is defined by

$$C_{id} = \bigcup_{\alpha \in [0,1]} {}_{\alpha}C_{id} \tag{2.45}$$

where

$${}^{\alpha}[C_{id}] = \frac{\sum_{k=1}^{N} ({}^{\alpha}[u_{ki}])^m \, ({}^{\alpha}[X_{kd}])}{\sum_{k=1}^{N} ({}^{\alpha}[u_{ki}])^m} \tag{2.46}$$

and $d = 1, \ldots, p$.

However, the KM algorithms by Karnik and Mendel [12] are used to compute interval weighted averages in the LFCM prototype update as an alternative to (2.46). We elucidate the KM algorithms for updating an endpoint of a cluster prototype in Algorithm 2.

Algorithm 2: Prototype update using the KM algorithm

Input: *endpoint*, ${}^{\alpha}[X_k^{(d)}]$, and $({}^{\alpha}[u_{ki}])^m, \forall k$
Output: An endpoint value of ${}^{\alpha}\left[\vec{C}_i^{(d)}\right]$

for $j \leftarrow 1$ **to** n **do**
  $x_j = \begin{cases} \vec{X}_{j[l]}^{(k)} & \text{if } \textit{endpoint} \text{ is “lower”} \\ \vec{X}_{j[r]}^{(k)} & \text{if } \textit{endpoint} \text{ is “upper”} \end{cases}$
  $w_j = \text{mid}(u_j)$
  $c_j = u_{j[l]}$
  $d_j = u_{j[r]}$
end
$x_{\text{sorted}}, x_{\text{index}} = \text{sort}([x_1, x_2, \ldots, x_j, \ldots, x_n])$
$\bar{x} = (\sum_{j=1}^{N} w_j^m x_j)/(\sum_{j=1}^{N} w_j^m)$
for $t \leftarrow 1$ **to** $n+1$ **do**
  Find k where $x_k \le \bar{x} \le x_{k+1}$
  $left\_indices = x_{\text{index}}[1:k]$
  $right\_indices = x_{\text{index}}[k+1:n]$
  if *endpoint is “lower”* **then**
    $w[left\_indices] = d[left\_indices]$
    $w[right\_indices] = c[right\_indices]$
  else if *endpoint is “upper”* **then**
    $w[left\_indices] = c[left\_indices]$
    $w[right\_indices] = d[right\_indices]$
  `// Termination`
  $\bar{x}_t = (\sum_{j=1}^{N} w_j^m x_j)/(\sum_{j=1}^{N} w_j^m)$
  if $|\bar{x} - \bar{x}_t| < \epsilon$ **then**
    break
  end
  $\bar{x} = \bar{x}_t$
end

2.3.4 Stopping Criteria

There are many ways that people use to terminate numerical clustering algorithms. However, it becomes more problematic with fuzzy numbers. For instance, how to determine when the cluster centers stabilize. One can be creative about the termination of the LFCM algorithm. The standard stopping criterion is based on the dissimilarity between the previous cluster prototypes and the updated cluster prototypes [7]. But, here we let the LFCM run for a certain number of iterations so that we can investigate other issues such as the membership spread problem and the uncertainty associated with the cluster centers.

2.4 Linguistic Possibilistic C-Means

The type-1 fuzzy C-means (FCM) was extended to the type-1 possibilistic C-means (PCM) by Krishnapuram and Keller [13] back in 1993. The idea of PCM is that the memberships u_{ji} does not need to sum to unity, for all i as they interpret the memberships as typicalities. So, we use the terms membership and typicality interchangeably for PCM and LFCM. Thus, they relax the probabilistic constraint defined in (2.32), and the only condition needed is that

$$u_{ji} \in [0,1], \forall j = 1, \ldots, N \text{ and } \forall i = 1, \ldots, C. \tag{2.47}$$

The objective function of PCM we need to minimize is

$$J(B, U; X) = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ji})^m (d_{ji})^2 + \sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} (1 - u_{ji})^m, \tag{2.48}$$

where η_i is a suitable positive real number. Now, we can see that there are two components in the criterion function. The former is exactly the same as the FCM objective function. The latter is there to keep the memberships as large as possible. Without the second component, PCM will likely to yield zero memberships.

The PCM membership equation is achieved by [13]

$$u_{ji} = \frac{1}{1 + \left(\frac{(d_{ji})^2}{\eta_i} \right)^{1/(m-1)}} \tag{2.49}$$

Notice that we now have one more C-tuple of parameters $\langle \eta_i \rangle$ involved. η_i is a suitable positive number associated with the cluster i. Its value is actually the squared distance where the membership

is 0.5. For instance, given $\eta_1 = 2$, $u_{1,1} = 0.5$ if $(d_{1,1})^2 = \eta_1 = 2$. Hence, η_i are used to determine the cluster sizes. However, finding good values for the parameters η_i is another big problem. The authors suggest that $\langle \eta_i \rangle$ can be estimated using

$$\eta_i = K \frac{\sum_{j=1}^{N} (u_{ji})^m (d_{ji})^2}{\sum_{j=1}^{N} (u_{ji})^m} \tag{2.50}$$

where K is usually set to 1.

Algorithm 3 summarizes the LPCM clustering algorithm that involves the same steps as in LFCM but we need to estimate η_i and the membership update equations are different. In terms of the cluster prototypes, they are updated using Algorithm 2 in the same manner as the LFCM.

Algorithm 3: Linguistic possibilistic C-means (LPCM)

Input: The number of clusters C, the dataset $\mathbf{X}$
Output: The C-tuple of prototypes B, the C-partition matrix U
1 Initialize cluster prototypes $\vec{C}_i$, $i = 1, \ldots, C$
2 Initialize C-partition matrix U using the LFCM (optional)
3 Set η_i
4 **repeat**
5 Compute fuzzy Euclidean distance $d_{ji}, \forall j, i$ using (2.37) and (2.39)
6 Compute membership $u_{ji}, \forall j, i$ using (2.54) and (2.55)
7 Update cluster prototype $\vec{C}_i, \forall i$ using Algorithm 2
8 **until** *Convergence*;

We can also estimate the bandwidths η_i in the form of fuzzy numbers. The extension principle along with the decomposition theorem are used on Equation 2.50, so we arrive at

$$\eta_i = \bigcup_{\alpha \in [0,1]} {}^{\alpha}[\eta_i] \tag{2.51}$$

where

$${}^{\alpha}[\eta_i] = K \frac{\sum_{j=1}^{N} ({}^{\alpha}[u_{ji}])^m ({}^{\alpha}[d_{ji}^2])}{\sum_{j=1}^{N} ({}^{\alpha}[u_{ji}])^m}. \tag{2.52}$$

Again, we can make use of the KM algorithms to calculate this equation as it is in the form of weighted average. But in the end, the LPCM treats η_i as real numbers so we need to apply some defuzzification techniques on η_i if they are computed in the form of fuzzy numbers.

2.4.1 LPCM Memberships

We can extend (2.49) using the extension principle, thus the expression used to update the LPCM membership of $\vec{X}_j$ in cluster i becomes

$$u_{ji} = \frac{1}{1 + \left(\frac{(d_{ji})^2}{\eta_i}\right)^{1/(m-1)}}. \tag{2.53}$$

Now, we do not need the distances to other clusters, except d_{ji}, to update the membership u_{ji}. So, if there are noise points present in the dataset, they will have low memberships (typicalities) as they should be far away from the clusters. In the framework of LPCM, we can apply the extension principle to transform Equation 2.53 into the following:

$${}^{\alpha}[u_{ji}] = \frac{1}{1 + \left(\frac{{}^{\alpha}[d_{ji}^2]}{\eta_i}\right)^{1/(m-1)}} \tag{2.54}$$

After that, the decomposition theorem is used to build up the membership u_{ji} as a fuzzy number. So, we arrive at

$$u_{ji} = \bigcup_{\alpha \in [0,1]} {}_{\alpha}[u_{ji}]. \tag{2.55}$$

It is really important to note that the parameters η_i used in the LPCM clustering algorithm (in (2.54)) for our applications are positive real numbers, not fuzzy numbers. So, if η_i are estimated using (2.51) and (2.52), we need to perform defuzzification to convert fuzzy numbers back to regular numbers.

Example 2.5: Computing LPCM memberships.

Given $\alpha = 0.5$, $m = 4$, $\eta_1 = \eta_2 = 10$, ${}^{\alpha}\left[(d_{5,1})^2\right] = [0, 80]$ and ${}^{\alpha}\left[(d_{5,2})^2\right] = [576, 1049]$.

$$
\begin{aligned}
\text{The membership } {}^{\alpha}[u_{5,1}] &= \frac{1}{1 + \left(\frac{d_{5,1}^2}{\eta_1}\right)^{1/(m-1)}} \\
&= \frac{1}{1 + ([0, 80]/10)^{1/(4-1)}} \\
&= \frac{1}{1 + [0, 8]^{1/3}} \\
&= \frac{1}{1 + [0, 2]} \\
&= \frac{1}{[1, 3]} \\
&= [1, 1] \times \left[\frac{1}{3}, \frac{1}{1}\right] \\
&= [0.33, 1.00]
\end{aligned}
$$

$$
\begin{aligned}
\text{The membership } {}^{\alpha}[u_{5,2}] &= \frac{1}{1 + \left(\frac{d_{5,2}^2}{\eta_2}\right)^{1/(m-1)}} \\
&= \frac{1}{1 + ([576, 1049]/10)^{1/(4-1)}} \\
&= \frac{1}{1 + [57.6, 104.9]^{1/3}} \\
&= \frac{1}{1 + [3.86, 4.72]} \\
&= \frac{1}{[4.86, 5.72]} \\
&= [1, 1] \times \left[\frac{1}{5.72}, \frac{1}{4.86}\right] \\
&= [0.18, 0.21]
\end{aligned}
$$

Example 2.6: Computing LPCM memberships.

Given $\alpha = 0.5$, $m = 2$, $\eta_1 = \eta_2 = 10$, ${}^{\alpha}\left[(d_{5,1})^2\right] = [0, 80]$ and ${}^{\alpha}\left[(d_{5,2})^2\right] = [576, 1049]$.

$$
\begin{aligned}
\text{The membership } {}^{\alpha}[u_{5,1}] &= \frac{1}{1 + \left(\frac{d_{5,1}^2}{\eta_1}\right)^{1/(m-1)}} \\
&= \frac{1}{1 + ([0, 80]/10)^{1/(2-1)}} \\
&= \frac{1}{1 + [0, 8]} \\
&= \frac{1}{[1, 9]} \\
&= [1, 1] \times \left[\frac{1}{9}, \frac{1}{1}\right] \\
&= [0.11, 1.00]
\end{aligned}
$$

$$
\begin{aligned}
\text{The membership } {}^{\alpha}[u_{5,2}] &= \frac{1}{1 + \left(\frac{d_{5,2}^2}{\eta_2}\right)^{1/(m-1)}} \\
&= \frac{1}{1 + ([576, 1049]/10)^{1/(2-1)}} \\
&= \frac{1}{1 + [57.6, 104.9]} \\
&= \frac{1}{[58.6, 105.9]} \\
&= [1, 1] \times \left[\frac{1}{105.9}, \frac{1}{58.6}\right] \\
&= [0.01, 0.02]
\end{aligned}
$$

Example 2.7: Computing LPCM memberships.

Given $\alpha = 0.5$, $m = 1.25$, $\eta_1 = \eta_2 = 10$, ${}^{\alpha}\left[(d_{5,1})^2\right] = [0, 80]$ and ${}^{\alpha}\left[(d_{5,2})^2\right] = [576, 1049]$.

$$
\begin{aligned}
\text{The membership } {}^{\alpha}[u_{5,1}] &= \frac{1}{1 + \left(\frac{d_{5,1}^2}{\eta_1}\right)^{1/(m-1)}} \\
&= \frac{1}{1 + ([0, 80]/10)^{1/(1.25-1)}} \\
&= \frac{1}{1 + [0, 8]^4} \\
&= \frac{1}{[1, 4097]} \\
&= [1, 1] \times \left[\frac{1}{4097}, \frac{1}{1}\right] \\
&= [0.00, 1.00]
\end{aligned}
$$

$$
\begin{aligned}
\text{The membership } {}^{\alpha}[u_{5,2}] &= \frac{1}{1 + \left(\frac{d_{5,2}^2}{\eta_2}\right)^{1/(m-1)}} \\
&= \frac{1}{1 + ([576, 1049]/10)^{1/(1.25-1)}} \\
&= \frac{1}{1 + [57.6, 104.9]^4} \\
&= [0.00, 0.00]
\end{aligned}
$$

According to the three examples above, if we consider $u_{5,1}$ with different m values, we can see the intervals as depicted in Figure 2.9. It is worthwhile to notice that, there is no obvious relation between the fuzzifier m and the uncertainty in terms of intervals for LPCM. This needs a more careful study.

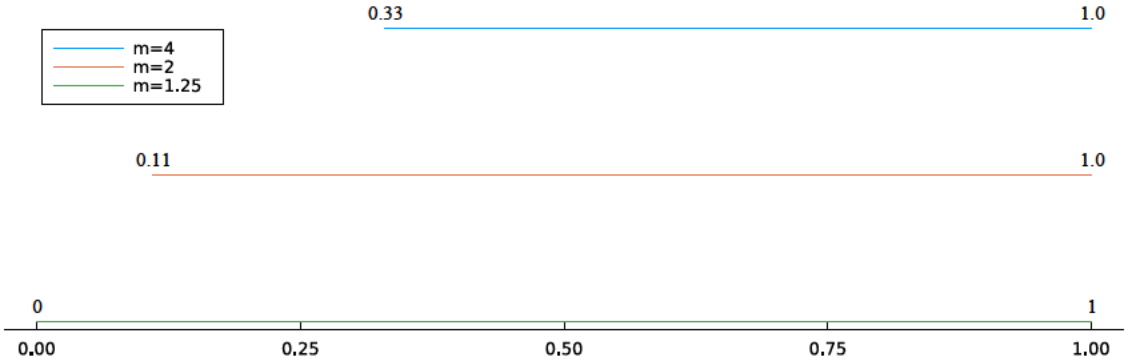


Figure 2.9: LPCM $u_{5,1}$ with different m values.

Chapter 3

TYPE-2 FUZZY CLUSTERING

3.1 Linguistic Fuzzy C-Means

3.1.1 LFCM Clustering

To investigate the LFCM, we will take full advantage of synthetic dataset that provides us the predictable behavior in a controlled manner. Therefore, we choose the fifteen-point butterfly dataset that was originally created by Ruspini [14], and index them as displayed in Table 3.1.

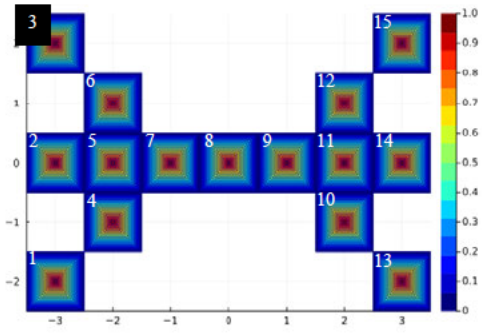
Table 3.1: Fifteen-point butterfly dataset.

Index	x	y
1	-3	-2
2	-3	0
3	-3	2
4	-2	-1
5	-2	0
6	-2	1
7	-1	0
8	0	0
9	1	0
10	2	-1
11	2	0
12	2	1
13	3	-2
14	3	0
15	3	2

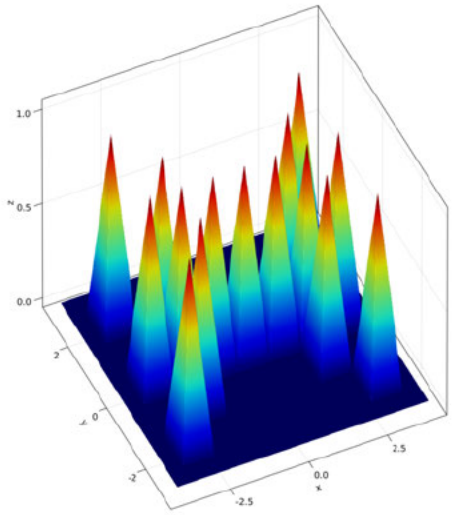
The patterns 1–7 are symmetrical to the patterns 9–15. So, we expect to see two clusters, each for one wing. The pattern $\vec{X}_8$, at $[0, 0]$, is the bridge point that is located in the middle between the two clusters. That means the prototypical points should be around the 5^{th} and the 11^{th} patterns.

The fuzzified butterfly dataset

To make the dataset work with the type-2 framework, we need to convert all real vectors into vectors of fuzzy numbers. Here, we choose to fuzzify each real number with triangular fuzzy number with $w = 0.5$, i.e., the width is 0.5 as defined in Section 2.1. Hence, each pattern becomes a 2-dimensional fuzzy vector. For instance, $\vec{X}_5 = \langle$"about -2", "about 0"$\rangle$. Figure 3.1 displays the fuzzy vectors. Since we fuzzify every real number in each dimension to be a triangular fuzzy number, a 2D fuzzy vector becomes a pyramid in 3D, see Figure 3.1b, where the third dimension represents the degree of membership. So, from hereon, we refer to the fuzzifier butterfly dataset as just the butterfly dataset.



(a) The top-down contour plot.



(b) The 3D plot.

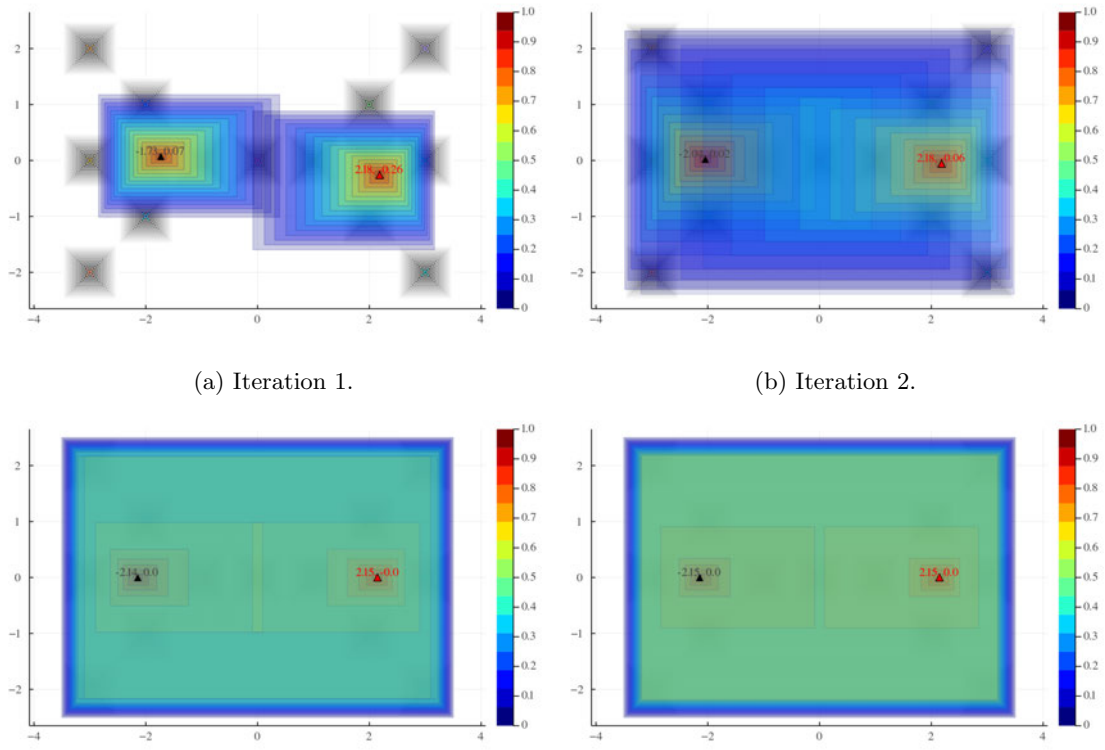
Figure 3.1: The visualization of the fuzzified butterfly dataset. There are 15 crisp points that are converted into 15 fuzzy vectors. Triangular fuzzy number is used in each dimension. Each of the fuzzy vectors has its index (the white number) indicated in the top-left corner.

The LFCM with $m = 2$

We perform the LFCM on the fuzzified butterfly dataset. The two initial cluster centers are manually chosen to be the 7^{th} and the 10^{th} patterns, respectively. The fuzzifier m is fixed to be 2 since it gives us the memberships and prototypes that are not too hard or not too fuzzy.

We can see the clustering results from Figure 3.2. The LFCM updates the cluster centers at iteration 1 to be as shown in Figure 3.2a. The colored pyramids represent the cluster centers (there are two pyramids in this case), whereas the black pyramids depict the butterfly patterns. The cluster centers are pyramids in 3D as a consequence of representing patterns with triangular fuzzy numbers, but here they are visualized in the form of a contour plot. Even at iteration 1, we already observe

the lower level cuts of the cluster centers that span several nearby patterns. After that, the cluster centers spread out and cover the whole space at iteration 2. Without a doubt, the uncertainty is even larger at iteration 5 when the convergence is assumed to be reached in this case. Here, a problem of the LFCM is that its uncertainty increases rapidly, not only in the memberships, but also in the cluster centers. Although the length of the base of a triangular fuzzy number used in this work is only 1.0, i.e., the input fuzzy vectors are not too fuzzy, the uncertainty is out of control.



(a) Iteration 1.

(b) Iteration 2.

(c) Iteration 5.

(d) Iteration 10.

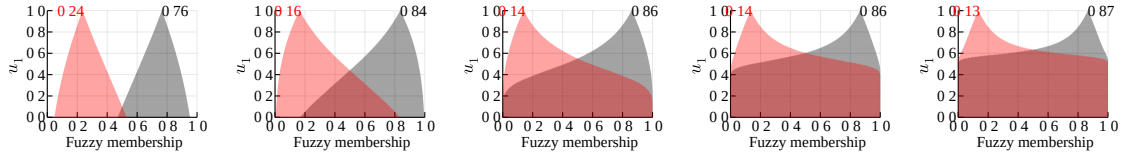
Figure 3.2: The resulting cluster centers (colored contours) from running the LFCM with $m = 2$ on the fuzzified butterfly dataset (black contours) at different iterations.

It is also important to note that the 1-cuts (the cores) of fuzzy numbers provide intervals that are degenerate, that is the peaks are not intervals but reduced down to real numbers. This is another effect of fuzzifying patterns in each dimension with triangular fuzzy numbers. Therefore, they provide the same result as the regular FCM if we take only the results from the cores. As you can see from Figure 3.2, the triangle labels illustrates the type-1 FCM cluster centers at different iterations. The cores of the cluster centers are shifting toward what we expect from the type-1 FCM. At iteration 5, the two cluster centers are at about $(-2.15, 0.00)$ and $(2.14, 0.00)$, respectively. If we cross-check them against the cluster centers from the FCM, we will get the same results. This confirms that the

LFCM (type-2) generalizes the FCM (type-1) when triangular fuzzy numbers are used in fuzzification. We note that if other methods are used to represent the features as fuzzy numbers, this extension will not be obvious. We use triangles here because they are easy to visualize and to help our intuition.

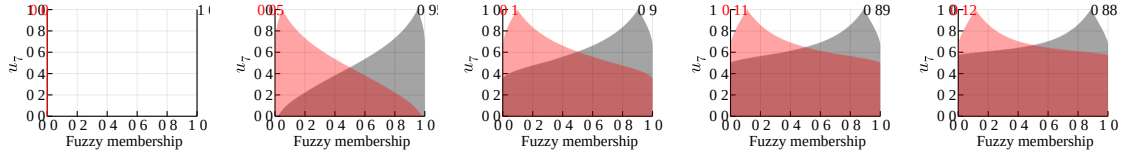
The supports of the cluster centers almost cover the whole space already at iteration 2. Although the cores of the cluster centers stop changing after iteration 5, the lower level cuts of the cluster centers still change. If we let it run for more iteration, the cluster centers at iteration 10 are even more uncertain as shown in Figure 3.2d.

The problem of membership spread can be seen clearly through the plots of membership functions. Hence, we show graphically the membership grades—which of course are fuzzy numbers—u_1, u_7, u_8 and u_{10} in Figure 3.3, Figure 3.4, Figure 3.5 and Figure 3.6, respectively. At iterations 2 to 5, the membership functions quickly spread out, that is, the uncertainty grows very fast.



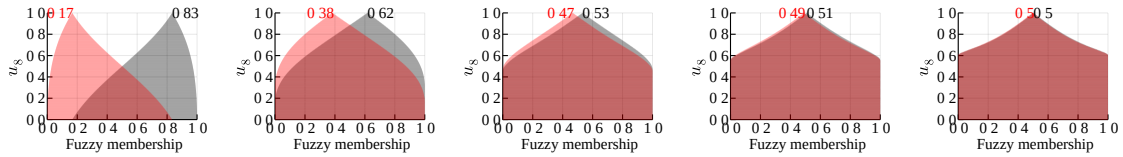
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.3: The membership functions $u_{1,1}$ and $u_{1,2}$ from running the LFCM with $m = 2$ on the butterfly dataset at different iterations. The black functions represent $u_{1,1}$, whereas the red functions represent $u_{1,2}$.



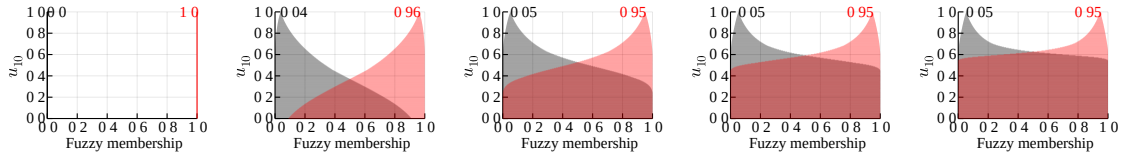
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.4: The membership functions $u_{7,1}$ and $u_{7,2}$ from running the LFCM with $m = 2$ on the butterfly dataset at different iterations. The black functions represent $u_{7,1}$, whereas the red functions represent $u_{7,2}$. They are both singleton fuzzy numbers at the beginning since the 7^{th} pattern is the initial cluster center.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.5: The membership functions $u_{8,1}$ and $u_{8,2}$ from running the LFCM with $m = 2$ on the butterfly dataset at different iterations. The black functions represent $u_{8,1}$, whereas the red functions represent $u_{8,2}$.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

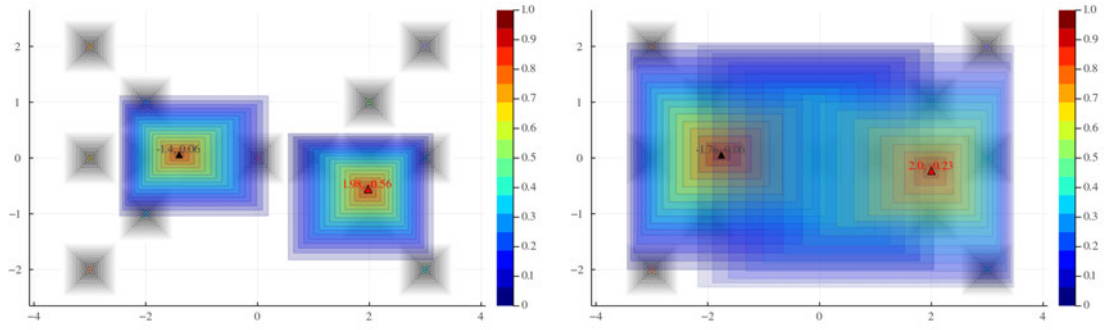
Figure 3.6: The membership functions $u_{10,1}$ and $u_{10,2}$ from running the LFCM with $m = 2$ on the butterfly dataset at different iterations. The black functions represent $u_{10,1}$, whereas the red functions represent $u_{10,2}$. They are both singleton fuzzy numbers at the beginning since the 10^{th} pattern is the initial cluster center.

Now consider the memberships of the pattern $\vec{X}_{10}$ in the two clusters, $u_{10,1}$ and $u_{10,2}$. Figure 3.6 depicts them at different iterations of the LFCM on this dataset. Since $\vec{X}_{10}$ is selected to be an initial cluster center, its $u_{10,1}$ and $u_{10,2}$ then are singleton fuzzy numbers (as defined in (2.41)) in the first iteration. After that, there is an obvious surge in uncertainty of the original membership functions. Starting at iteration 2 (Figure 3.6b), the uncertainty builds up rapidly. The supports of $u_{10,1}$ and $u_{10,2}$ almost fill up the interval $[0, 1]$. This is a phenomenon that causes the membership spread issue. They are even more uncertain as the iterations progress. After iteration 4, the memberships $u_{10,1}$ and $u_{10,2}$ for $\alpha \in [0.0, 0.6]$ are uninformative as shown in Figure 3.6d. The reason is the possible value of a grade of membership is in the range between 0 and 1, but our intervals at those α-cuts are also $[0, 1]$, i.e., the uncertainty is maximum.

Apart from the number of clusters C, the fuzzifier m is another hyperparameter that we need to pick. We will investigate the effects of different m values such as $m \in \{1.25, 2, 4\}$.

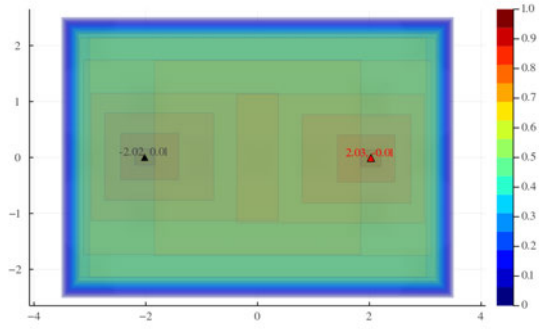
The LFCM with $m = 4$

Next, we set $m = 4$, meaning that we increase the fuzziness through the fuzzifier m that is utilized in computing both the memberships and the cluster centers. The clustering results at iteration 1, 2 and 5 are depicted in Figure 3.7.

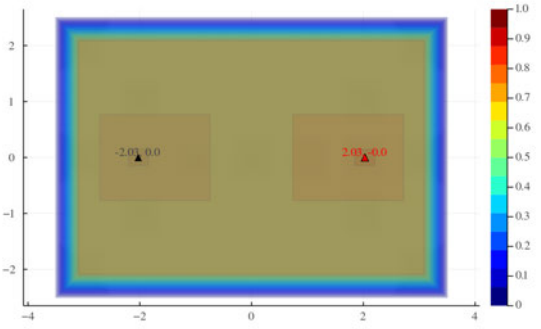


(a) Iteration 1.

(b) Iteration 2.



(c) Iteration 5.



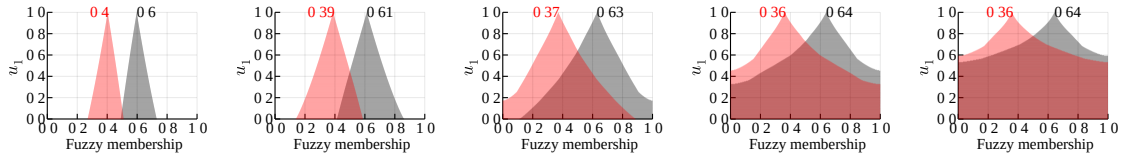
(d) Iteration 10.

Figure 3.7: The resulting cluster centers (colored contours) from running the LFCM with $m = 4$ on the fuzzified butterfly dataset (black contours) at different iterations.

At iteration 1, the cores of the cluster centers are at $(-1.73, 0.07)$ and $(2.18, -0.26)$, respectively. The LFCM with $m = 2$ seems to produce more uncertain cluster centers than the LFCM with $m = 4$. If we take a closer look at iteration 1, the LFCM with $m = 2$ has more noticeable uncertainty in the x-axis than the LFCM with $m = 4$, especially for the lower level cuts. At iteration 2, the uncertainty grows rapidly, but not as quickly as when $m = 2$, i.e., the lower α-cuts do not span the whole space completely. However, this is still out of control. The peaks of the final cluster centers are are $(-2.03, 0.00)$ and $(2.03, 0.00)$, respectively, at iteration 10. However, the lower level cuts of them contain more uncertainty than that of the LFCM with $m = 2$. Conclusively, setting $m = 4$ seems to produce less uncertain cluster centers in the beginning, but they have more uncertainty than that of $m = 2$ in the end.

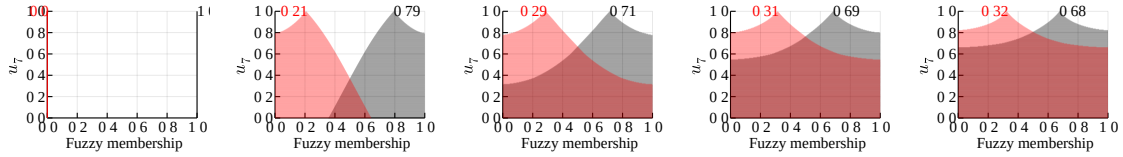
Next, let us consider the LFCM memberships u_1, u_7, u_8, and u_{10} which are plotted in Figures 3.8–3.11. The LFCM membership functions of the pattern $\vec{X}_1$, $u_{1,1}$ and $u_{1,2}$, with $m = 4$ at iterations 1–5 are illustrated in Figure 3.8. Comparing them with the ones with $m = 2$ (Figure 3.3), we observe that $m = 4$ gives us distinguishable skinnier fuzzy memberships than that of $m = 2$, i.e., less

uncertainty. The memberships (at least the peaks) are more closer to $1/C$, in this case, 0.5 for $m = 4$. However, at iteration 5, the level cuts below 0.6 also become too fuzzy similar to the case of $m = 2$.



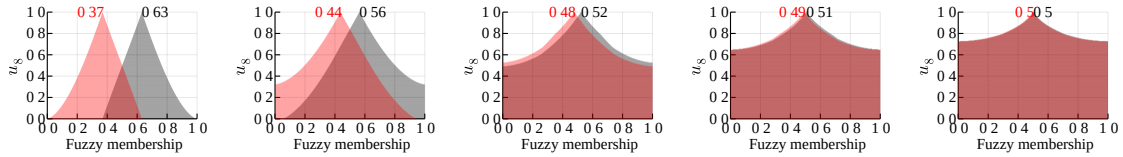
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.8: The membership functions $u_{1,1}$ and $u_{1,2}$ from running the LFCM with $m = 4$ on the butterfly dataset at different iterations. The black functions represent $u_{1,1}$, whereas the red functions represent $u_{1,2}$.



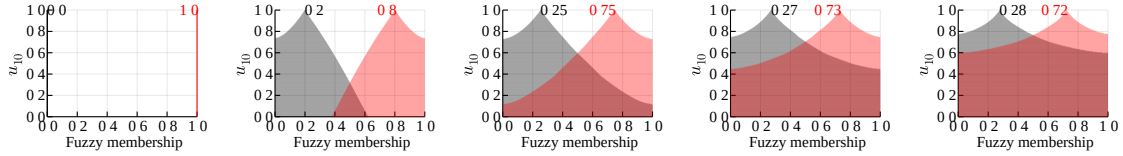
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.9: The membership functions $u_{7,1}$ and $u_{7,2}$ from running the LFCM with $m = 4$ on the butterfly dataset at different iterations. The black functions represent $u_{7,1}$, whereas the red functions represent $u_{7,2}$. They are both singleton fuzzy numbers at the beginning since the 7^{th} pattern is the initial cluster center.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.10: The membership functions $u_{8,1}$ and $u_{8,2}$ from running the LFCM with $m = 4$ on the butterfly dataset at different iterations. The black functions represent $u_{8,1}$, whereas the red functions represent $u_{8,2}$.

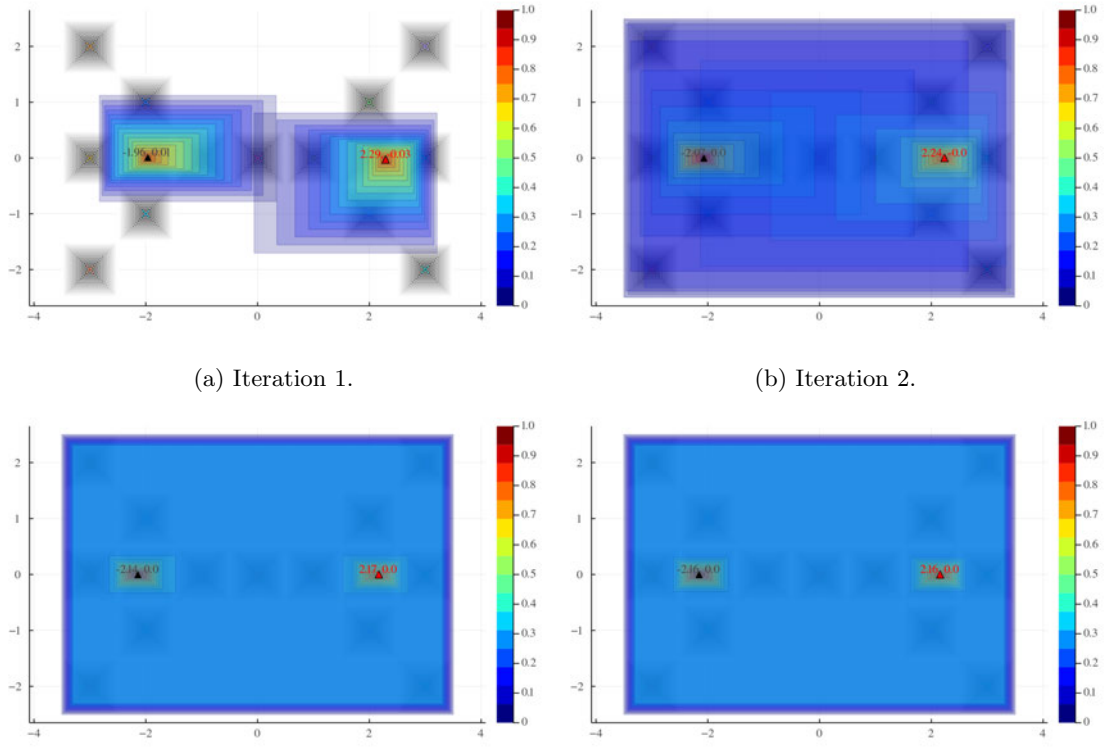


(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.11: The membership functions $u_{10,1}$ and $u_{10,2}$ from running the LFCM with $m = 4$ on the butterfly dataset at different iterations. The black functions represent $u_{10,1}$, whereas the red functions represent $u_{10,2}$. They are both singleton fuzzy numbers at the beginning since the 10^{th} pattern is the initial cluster center.

The LFCM with $m = 1.25$

On the flip side, if the fuzzifier m is reduced down to 1.25, the cluster centers from running the LFCM are portrayed in Figure 3.12. The cluster centers at iteration 1 show the most obvious difference of uncertainty between the x-axis and the y-axis. The peaks of the final cluster centers stop moving at iteration 7. However, the α-cuts where $\alpha \in [0.4, 1.0]$ of the cluster centers after iteration 7 are smaller (they are more certain) as the iterations progress, unlike that of the LFCM with $m = 2$ or $m = 4$.



(a) Iteration 1.

(b) Iteration 2.

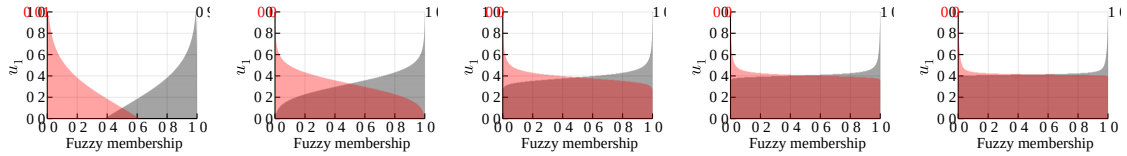
(c) Iteration 5.

(d) Iteration 10.

Figure 3.12: The resulting cluster centers (colored contours) from running the LFCM with $m = 1.25$ on the fuzzified butterfly dataset (black contours) at different iterations.

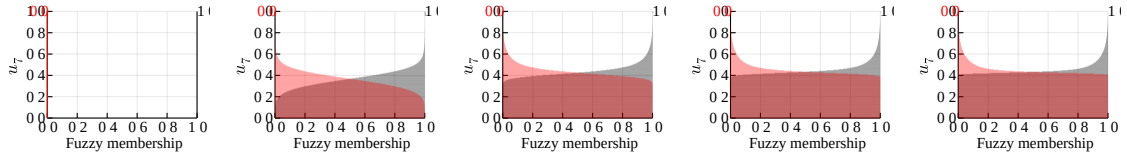
Figure 3.13, Figure 3.14, Figure 3.15 and Figure 3.16 illustrate the memberships u_1, u_7, u_8 and u_{10}, respectively, from the LFCM with $m = 1.25$. The cores of the memberships u_1, u_7 and u_{10} are strongly zero or unity, in contrast to the case with $m = 4$. The uncertainty skyrockets starting from iteration 2. Hence, $m = 1.25$ for the LFCM does not provide usefulness in terms of fuzzy memberships as the shape of them are strange, i.e., the lower level cuts ($\alpha < 0.4$) are uninformative, except the higher level cuts that provide the expected membership values (according to the FCM). So, this can confirm the LFCM converges to the LHCM as m approaches 1 from above (it is similar

to the case of the FCM).



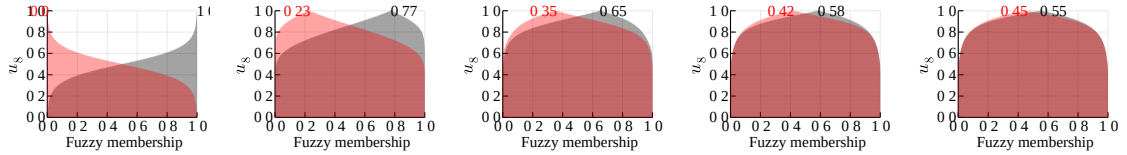
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.13: The membership functions $u_{1,1}$ and $u_{1,2}$ from running the LFCM with $m = 1.25$ on the butterfly dataset at different iterations. The black functions represent $u_{1,1}$, whereas the red functions represent $u_{1,2}$.



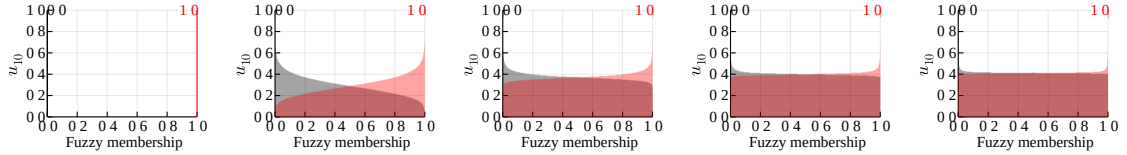
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.14: The membership functions $u_{7,1}$ and $u_{7,2}$ from running the LFCM with $m = 1.25$ on the butterfly dataset at different iterations. The black functions represent $u_{7,1}$, whereas the red functions represent $u_{7,2}$. They are both singleton fuzzy numbers at the beginning since the 7^{th} pattern is the initial cluster center.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.15: The membership functions $u_{8,1}$ and $u_{8,2}$ from running the LFCM with $m = 1.25$ on the butterfly dataset at different iterations. The black functions represent $u_{8,1}$, whereas the red functions represent $u_{8,2}$.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.16: The membership functions $u_{10,1}$ and $u_{10,2}$ from running the LFCM with $m = 1.25$ on the butterfly dataset at different iterations. The black functions represent $u_{10,1}$, whereas the red functions represent $u_{10,2}$. They are both singleton fuzzy numbers at the beginning since the 10^{th} pattern is the initial cluster center.

Figure 3.17 shows the U-uncertainty values of the first cluster prototype, $U(\vec{C}_1)$, from the LFCM

with different m values. One could notice that as m is larger, the uncertainty of $\vec{C}_1$ increases. Also, if we use the uncertainty of cluster centers to be one of the criteria, the bigger m tends to have a slower convergence.

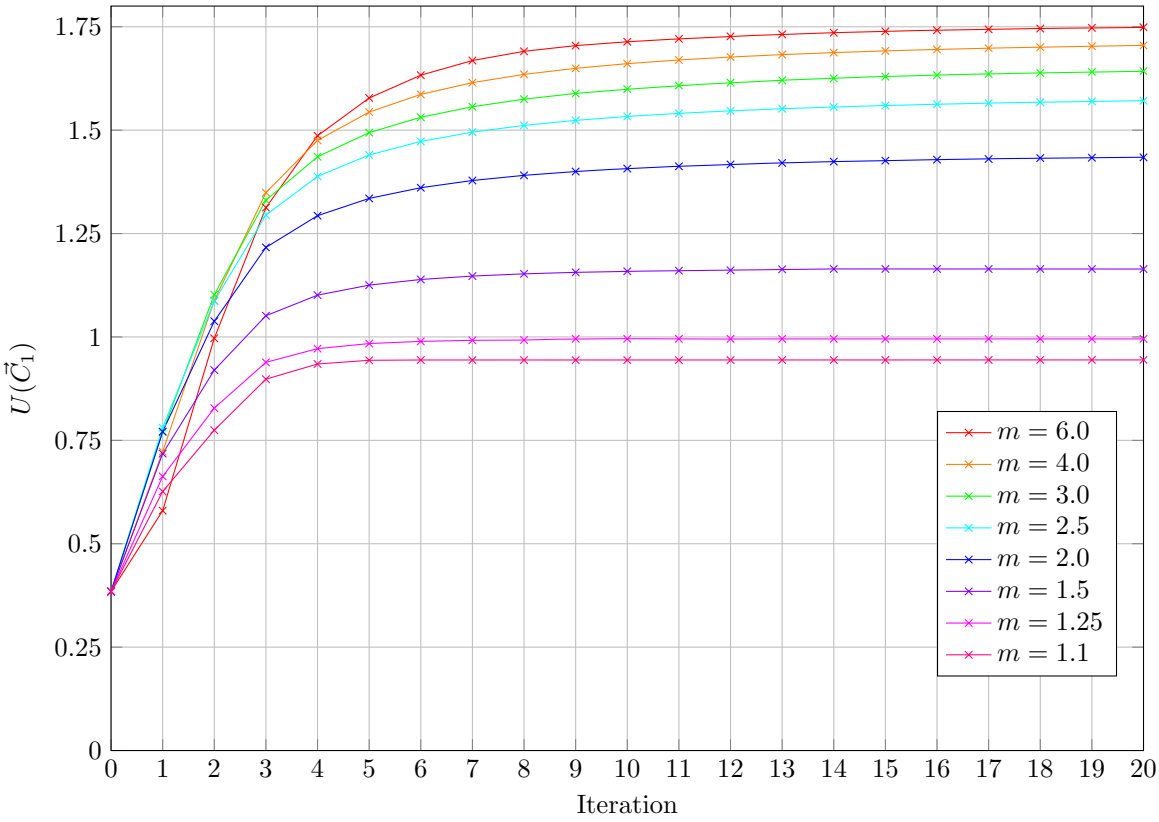


Figure 3.17: Plot of U-uncertainty values of the first cluster center, $U(\vec{C}_1)$, from the LFCM with different m values.

3.2 Dampening Approaches

As stated earlier, we found the uncertainty problem, both in the memberships and the corresponding cluster centers, in the type-2 FCM. Thus, it is possible to mitigate the problem by applying some dampening approaches to the membership functions. Here, we discuss three different dampening approaches to keep the uncertainty under control including 1) vertical cut dampening, 2) linear dampening, and 3) reflection dampening.

3.2.1 Vertical Cut Dampening

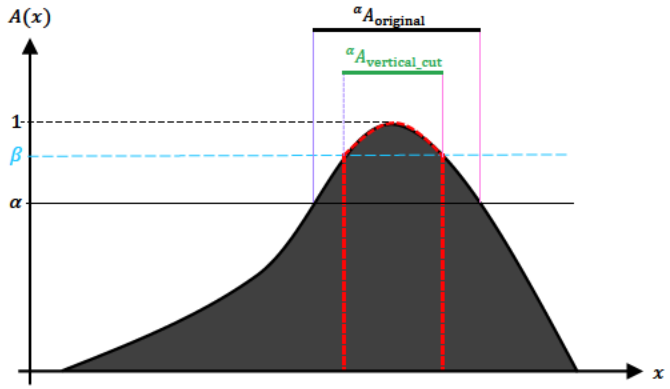


Figure 3.18: The vertical cut dampening method. All the level cuts below β duplicate the interval at β which is denoted by the blue solid line. The resulting membership function is depicted with the red dashed line.

Vertical cut dampening originates from the fact that the shapes of membership functions (fuzzy numbers) tend to be broad and spread out at the lower level cuts. Thus, the most straightforward way to trim membership functions is to cut them vertically. As we make use of α-cuts, we replace the intervals of a membership function at lower α-cuts with the specific interval at β-cut. This can be mathematically defined as

$$ {}^{\alpha}A(x) = {}^{\beta}A(x) \tag{3.1} $$

where $\alpha \in (0, \beta)$ for some $\beta \in (0, 1)$.

3.2.2 Linear Dampening

Following the characteristic of vertical cut dampening, we perceive that it dampens a membership function too much most of the time. The linear dampening is devised to simplify the shape of a membership function using straight lines. We utilize two lines from the peak to the two base endpoints which will result in a triangular function. The steps are straightforward as one needs to compute the left and right slopes, and then dampen them to some degree.

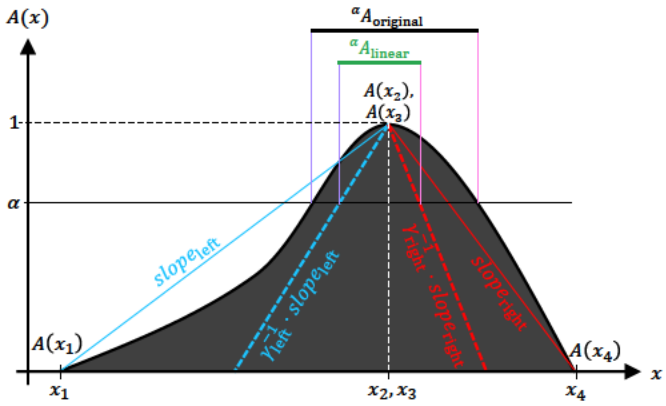


Figure 3.19: The linear dampening method. The solid lines are the estimated slopes of the left and right sides with respect to the middle point x'. The dashed lines represent the dampened slopes using (3.3).

Let us define ${}^{0+}A = [x_1, x_4]$ be a strong α-cut when $\alpha = 0$. This gives us a closed interval that is the support of fuzzy number A. Moreover, ${}^{1}A = [x_2, x_3]$ is a closed interval that represents the core of fuzzy number A. The middle point at the core the can be computed with

$$x' = \text{mid}\left({}^{1}A\right). \tag{3.2}$$

Since we are operating with normal, convex fuzzy numbers, then the membership value at x', i.e., $y' = A(x')$ is invariably unity. Each point x_i has the associated membership grade $y_i = A(x_i)$. Next, we will operate on one side at a time. Given a linear equation, we dampen the left slope and the right slope with the linear dampening ratios $\gamma_{[l]}$ and $\gamma_{[r]}$, respectively. We compute the left and right dampened slopes by the following equations:

$$slope_{[l]} = \frac{1}{\gamma_{[l]}} \cdot \frac{y' - y_1}{x' - x_1}, \tag{3.3a}$$

$$slope_{[r]} = \frac{1}{\gamma_{[r]}} \cdot \frac{y_4 - y'}{x_4 - x'} \tag{3.3b}$$

where $\gamma_{[l]}, \gamma_{[r]} \in (0, 1]$. Then, the corresponding intercepts of the left side and the right side are

$$b_{[l]} = y' - (slope_{[l]} \times x'), \tag{3.4a}$$

$$b_{[r]} = y' - (slope_{[r]} \times x'), \tag{3.4b}$$

respectively. Once we tool up all components of linear equations, the linear dampened membership

function at a given α-cut is modified using

$$ {}^{\alpha}A_{\text{linear}} = \left[\frac{\alpha - b_{[l]}}{slope_{[l]}}, \frac{\alpha - b_{[r]}}{slope_{[r]}}\right] = \left[A_{[l]}, A_{[r]}\right]. \tag{3.5} $$

To prevent the errors in the interval computation, we need to limit the endpoints, so the final left and right endpoints are:

$$ A'_{[l]} = \min\left(A_{[l]}, x'\right), \tag{3.6a} $$

$$ A'_{[r]} = \max\left(x', A_{[r]}\right), \tag{3.6b} $$

respectively, for all $\alpha \in (0, 1]$.

We would like to point out that, when $\gamma = 1.0$, it always yields a triangular fuzzy number which is a simplification of the original fuzzy number. Doing so helps reduce the computational complexity as one could invariably perform the LFCM clustering at only particular α-cuts (such as 0, 0.5, and 1.0), and interpolate the intervals in between.

3.2.3 Reflection Dampening

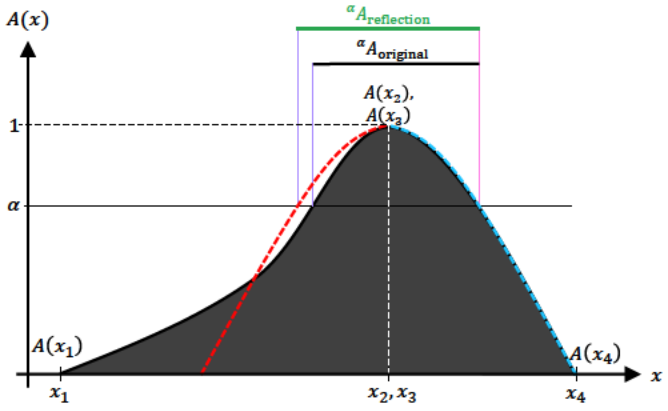


Figure 3.20: The reflection dampening method. The blue dashed line indicates the thinner side of the membership function. The red dashed line shows the reflection of the thinner side around x'.

The essence of reflection dampening is, not only shrinking the uncertainty that grows too fast, but also preserving the original shape of the fuzzy number to some degree. We observed that membership functions tend to have one side longer than another (using the peak as the reference). This is what we call the membership spread. Thus, to apply reflection dampening, we need to find the thinner side and then use the width between the peak to that side to reflect to the other side. The use of

α-cuts for finding the support and the core is as described above. Therefore, these steps are defined mathematically as:

$$ {}^{\alpha}A_{\text{reflection}} = \begin{cases} [a_\alpha, 2x' - a_\alpha] & \text{if } x' - x_1 < x_3 - x' \\ [2x' - b_\alpha, b_\alpha] & \text{otherwise} \end{cases} \tag{3.7} $$

where $[a_\alpha, b_\alpha]$ is the closed interval from the α-cut of the original membership function. We also need (3.6) to handle the edge cases.

3.2.4 Comparison Between Dampening Approaches

For the comparison sake, we perform the LFCM on the fuzzified butterfly dataset with the dampening approaches using the same setting as described earlier with the fixed $m = 2$. We provide a comparison between the membership functions with and without the dampening methods. Figure 3.21 displays the membership functions $u_{8,1}$ (black) and $u_{8,2}$ (red) with and without dampening methods. As the 8^{th} pattern is closer to the first cluster prototype, the membership function (fuzzy number) $u_{8,1}$ is then larger than $u_{8,2}$. The peaks of them sum to 1 due to the probabilistic constraint on the sum of memberships of the FCM algorithm. Again, this also confirms that the results are somewhat what we expect to see from the type-2 FCM based on the results from the type-1 FCM. We can clearly observe that the original membership functions are very wide, i.e., uncertainty is large. But after employing any proposed dampening methods, we trade in some information for the lesser uncertainty. This is the direct effect of the dampening approaches on the membership functions. As mentioned before, the original membership functions tend to spread on one side. Therefore, we make use of this fact to apply the reflection dampening to reflect the thinner side and also preserve the original shape of the membership function. Among all three proposed dampening approaches, reflection dampening seems to provide the best dampening method and is our preference. The cluster prototypes associated with the linear dampening with $\gamma = 0.5$ are still quite uncertain. The vertical cut dampening with $\beta = 0.8$—the simplest method to implement—applied to membership functions results in the least uncertainty, nonetheless it does not retain the original shapes of the membership functions. Hence, the reflection dampening method is the method of choice.

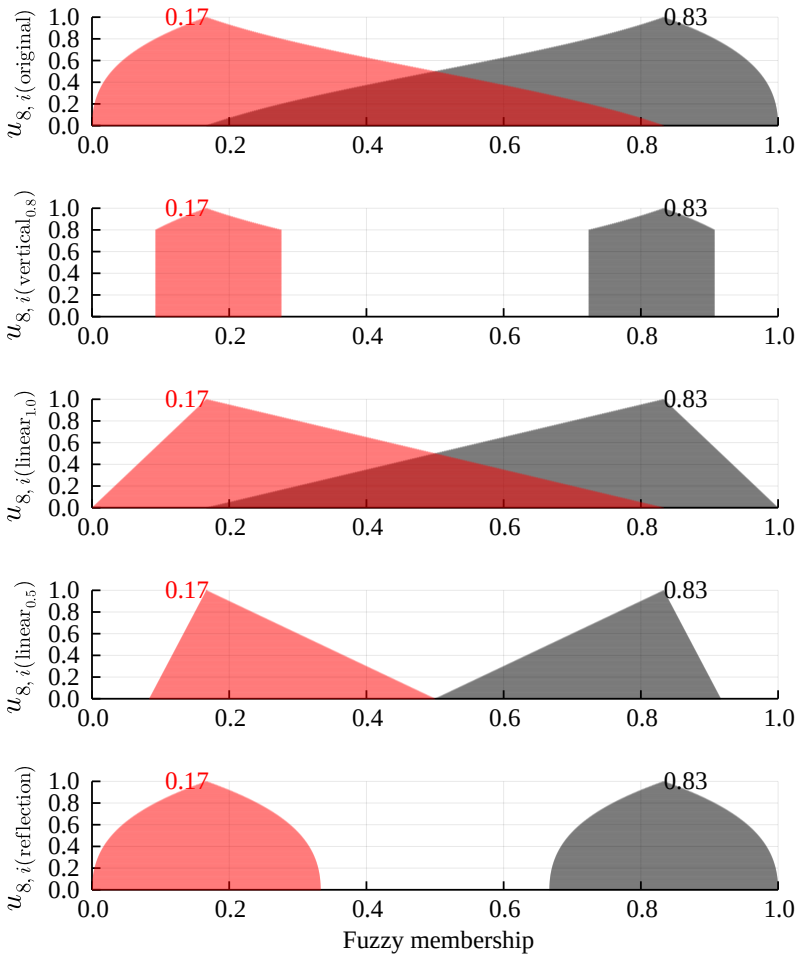
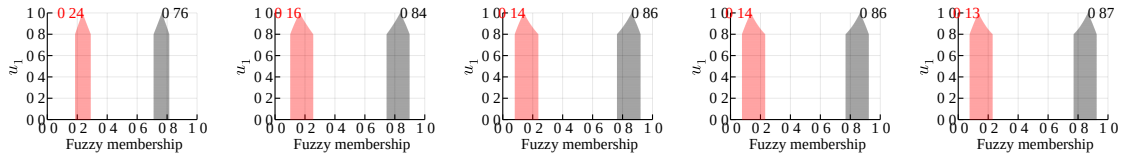


Figure 3.21: Comparing membership functions u_8 with different dampening approaches at iteration 1 of the LFCM. $u_{8,1}$ and $u_{8,2}$ are colored as black and red, respectively. From top to bottom: the original membership functions $u_{8,i}$, the vertical cut dampened $u_{8,i}$ with $\beta = 0.8$, the linear dampened $u_{8,i}$ with $\gamma = 1.0$, the liner dampened $u_{8,i}$ with $\gamma = 0.5$, and the reflection dampened $u_{8,i}$, where $i = 1, 2$.

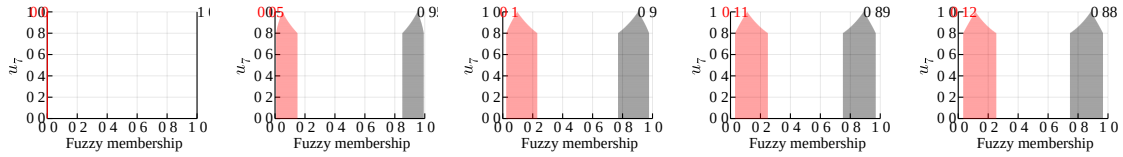
The LFCM with $m = 2$ and the vertical cut dampening ($\beta = 0.8$)

First, we set the hyperparameter of the vertical cut dampening β to be 0.8. It determines the levels that we will dampen in a membership function. Considering the vertical cut dampening approach, we can think of it as instead of taking the results from all level cuts, but only some level cuts that have $\alpha > \beta$. The reason is because the lower level cuts just duplicate the β-cut. Figure 3.22, Figure 3.23, Figure 3.24 and Figure 3.25 shows the memberships u_1, u_7, u_8 and u_{10} over iterations. We observe that the uncertainty of the memberships is much less than that of the original ones. This helps us see the values of the memberships in terms of fuzzy numbers easier.



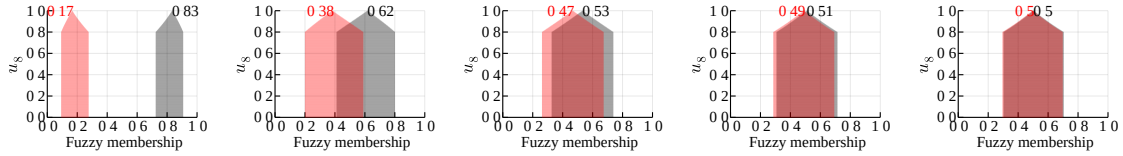
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.22: The vertical cut dampened LFCM membership functions $u_{1,1}$ and $u_{1,2}$ with $m = 2$ and $\beta = 0.8$ at different iterations. The black functions represent $u_{1,1}$, whereas the red functions represent $u_{1,2}$.



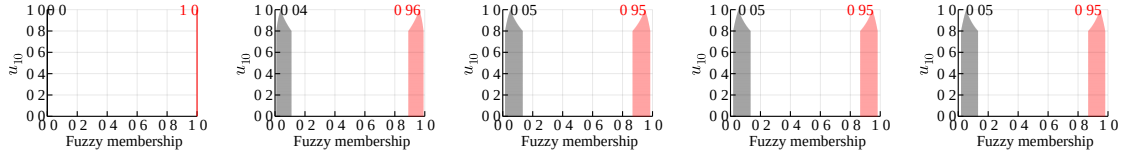
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.23: The vertical cut dampened LFCM membership functions $u_{7,1}$ and $u_{7,2}$ with $m = 2$ and $\beta = 0.8$ at different iterations. The black functions represent $u_{7,1}$, whereas the red functions represent $u_{7,2}$. They are both singleton fuzzy numbers at the beginning since the 7^{th} pattern is the initial cluster center.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.24: The vertical cut dampened LFCM membership functions $u_{8,1}$ and $u_{8,2}$ with $m = 2$ and $\beta = 0.8$ at different iterations. The black functions represent $u_{8,1}$, whereas the red functions represent $u_{8,2}$.

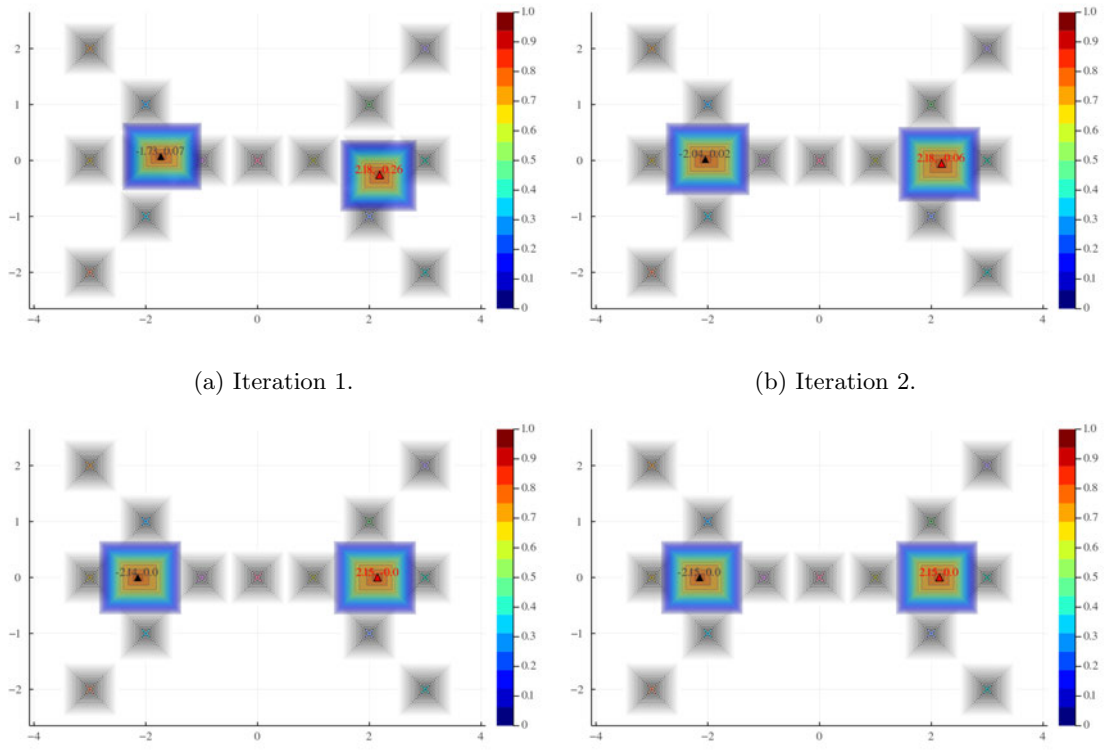


(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.25: The vertical cut dampened LFCM membership functions $u_{10,1}$ and $u_{10,2}$ with $m = 2$ and $\beta = 0.8$ at different iterations. The black functions represent $u_{10,1}$, whereas the red functions represent $u_{10,2}$. They are both singleton fuzzy numbers at the beginning since the 10^{th} pattern is the initial cluster center.

In terms of the resulting cluster centers calculated from the vertical cut dampened memberships with $\beta = 0.8$, they are illustrated in Figure 3.26. Applying the vertical cut dampening method also

indirectly affects the cluster centers as a result. Obviously, the peaks at different iterations are similar to the peaks from the memberships without dampening. As $\beta = 0.8$ is used here, the level cuts below 0.8 are identical to the 0.8-cuts. It can be viewed as a convenient method to eliminate the uncertainty of the lower level cuts that explodes into the space rapidly. This helps us to not get distracted by the uncertainty that is out of control. Also, after applying the vertical cut dampening in this case, we can clearly see that these two cluster centers are the reflection of each other.



(a) Iteration 1.

(b) Iteration 2.

(c) Iteration 5.

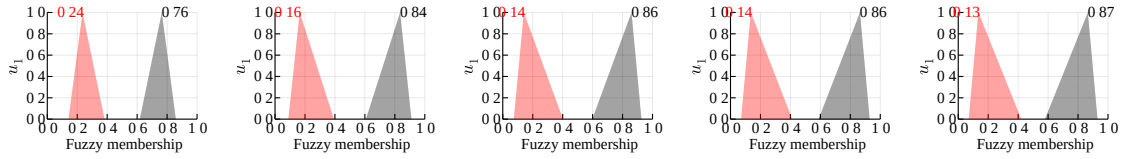
(d) Iteration 10.

Figure 3.26: The resulting cluster centers (colored contours) from running the LFCM with $m = 2$ on the fuzzified butterfly dataset (black contours) at different iterations. The memberships used to compute these cluster centers are dampened with the vertical cut dampening approach with $\beta = 0.8$.

The LFCM with $m = 2$ and the linear dampening ($\gamma = 0.5$)

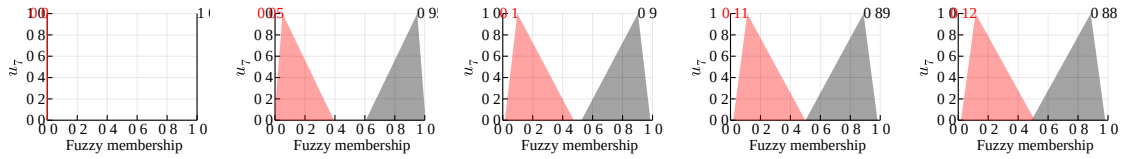
Next, let us consider the linear dampening approach that can be employed to mitigate the uncertainty problem. Again, the same setting used in the vertical cut dampening experiment is also used here but the dampening approach applied onto the memberships is the linear dampening with $\gamma = 0.5$. The dampened memberships u_1, u_7, u_8, u_{10} are depicted in Figure 3.27, Figure 3.28, Figure 3.29, Figure 3.30, respectively. One nice advantage of the linear dampening is that it always gives us the triangular memberships as shown in Figure 3.27. This comes from the fact that it operates on the

estimated left and right slopes of any given membership function. Doing so can help reduce the computational complexity because one could compute only a few α-cuts, for instance, $\alpha \in 0.0, 0.5, 1.0$, and interpolate the level cuts between them using linear equations.



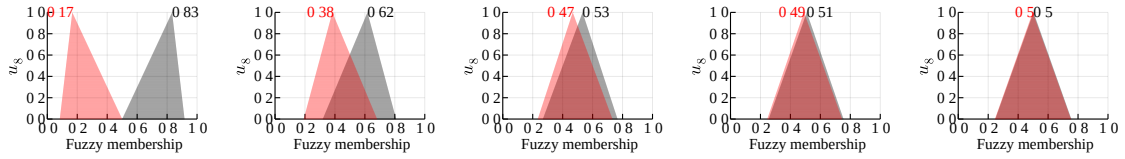
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.27: The linear dampened LFCM membership functions $u_{1,1}$ and $u_{1,2}$ with $m = 2$ and $\gamma = 0.5$ at different iterations. The black functions represent $u_{1,1}$, whereas the red functions represent $u_{1,2}$.



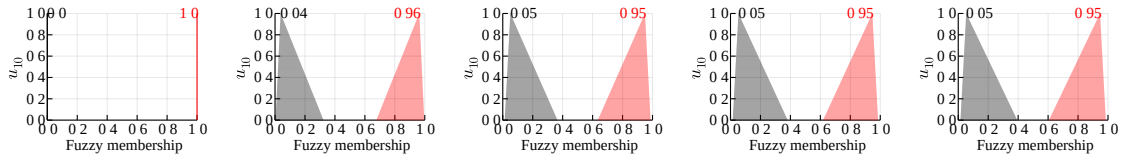
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.28: The linear dampened LFCM membership functions $u_{7,1}$ and $u_{7,2}$ with $m = 2$ and $\gamma = 0.5$ at different iterations. The black functions represent $u_{7,1}$, whereas the red functions represent $u_{7,2}$. They are both singleton fuzzy numbers at the beginning since the 7^{th} pattern is the initial cluster center.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

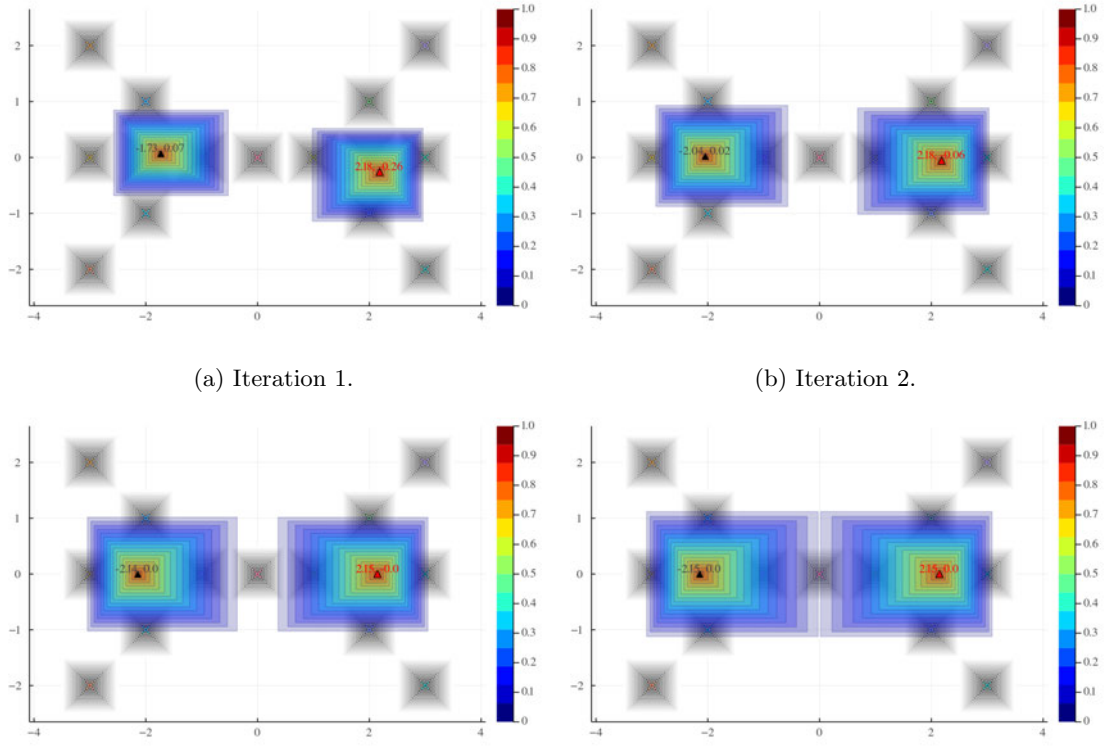
Figure 3.29: The linear dampened LFCM membership functions $u_{8,1}$ and $u_{8,2}$ with $m = 2$ and $\gamma = 0.5$ at different iterations. The black functions represent $u_{8,1}$, whereas the red functions represent $u_{8,2}$.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.30: The linear dampened LFCM membership functions $u_{10,1}$ and $u_{10,2}$ with $m = 2$ and $\gamma = 0.5$ at different iterations. The black functions represent $u_{10,1}$, whereas the red functions represent $u_{10,2}$. They are both singleton fuzzy numbers at the beginning since the 10^{th} pattern is the initial cluster center.

As portrayed in Figure 3.31, the cluster centers, computed from the linear dampened memberships with $\gamma = 0.5$, contain more uncertainty compared to the ones with vertical cut dampening, but contain less uncertainty compared to the original ones. The good here is that the uncertainty is suppressed to the point that the lower level cuts do not spread over the space, even at iteration 10.



(a) Iteration 1.

(b) Iteration 2.

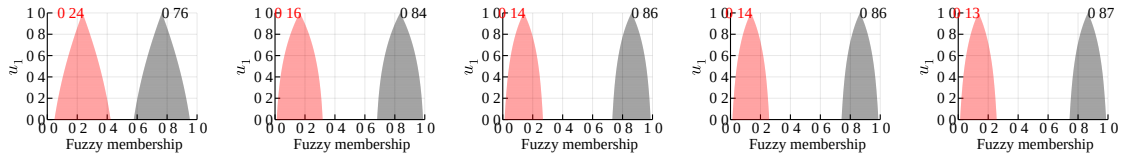
(c) Iteration 5.

(d) Iteration 10.

Figure 3.31: The resulting cluster centers (colored contours) from running the LFCM with $m = 2$ on the fuzzified butterfly dataset (black contours) at different iterations. The memberships used to compute these cluster centers are dampened with the linear dampening approach with $\gamma = 0.5$.

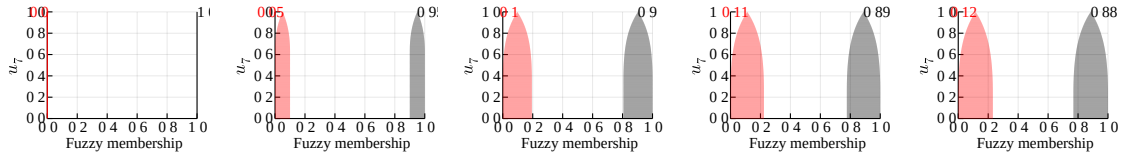
The LFCM with $m = 2$ and the reflection dampening

The last dampening approach—the reflection dampening—is employed here along with the same setting. Again, we would like to make a comparison between them. This dampening method does not require any hyperparameter to be set. The reflection dampened memberships u_1, u_7, u_8, u_{10} are depicted in Figure 3.32, Figure 3.33, Figure 3.34, Figure 3.35, respectively. The essence of this approach is to take the thinner side and reflect around the peak. It helps preserve the original shape of the membership to some extent. The cores are still the same as the original. The singleton fuzzy numbers passed onto the reflection dampening should not be modified as illustrated in Figure 3.33a and Figure 3.30a.



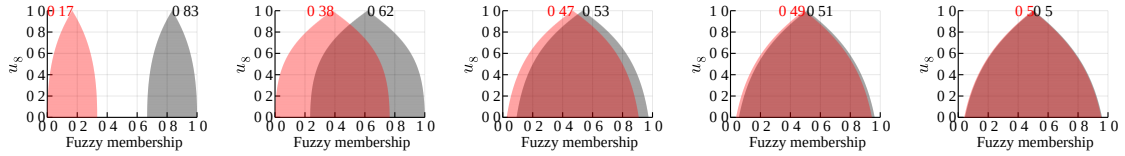
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.32: The reflection dampened LFCM membership functions $u_{1,1}$ and $u_{1,2}$ with $m = 2$ and $\gamma = 0.5$ at different iterations. The black functions represent $u_{1,1}$, whereas the red functions represent $u_{1,2}$.



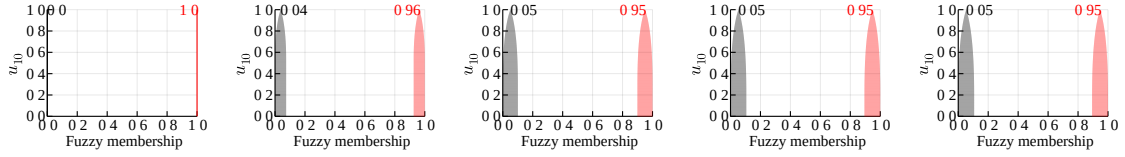
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.33: The reflection dampened LFCM membership functions $u_{7,1}$ and $u_{7,2}$ with $m = 2$ and $\gamma = 0.5$ at different iterations. The black functions represent $u_{7,1}$, whereas the red functions represent $u_{7,2}$. They are both singleton fuzzy numbers at the beginning since the 7^{th} pattern is the initial cluster center, and they are not affected by the reflection dampening.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.34: The reflection dampened LFCM membership functions $u_{8,1}$ and $u_{8,2}$ with $m = 2$ and $\gamma = 0.5$ at different iterations. The black functions represent $u_{8,1}$, whereas the red functions represent $u_{8,2}$.

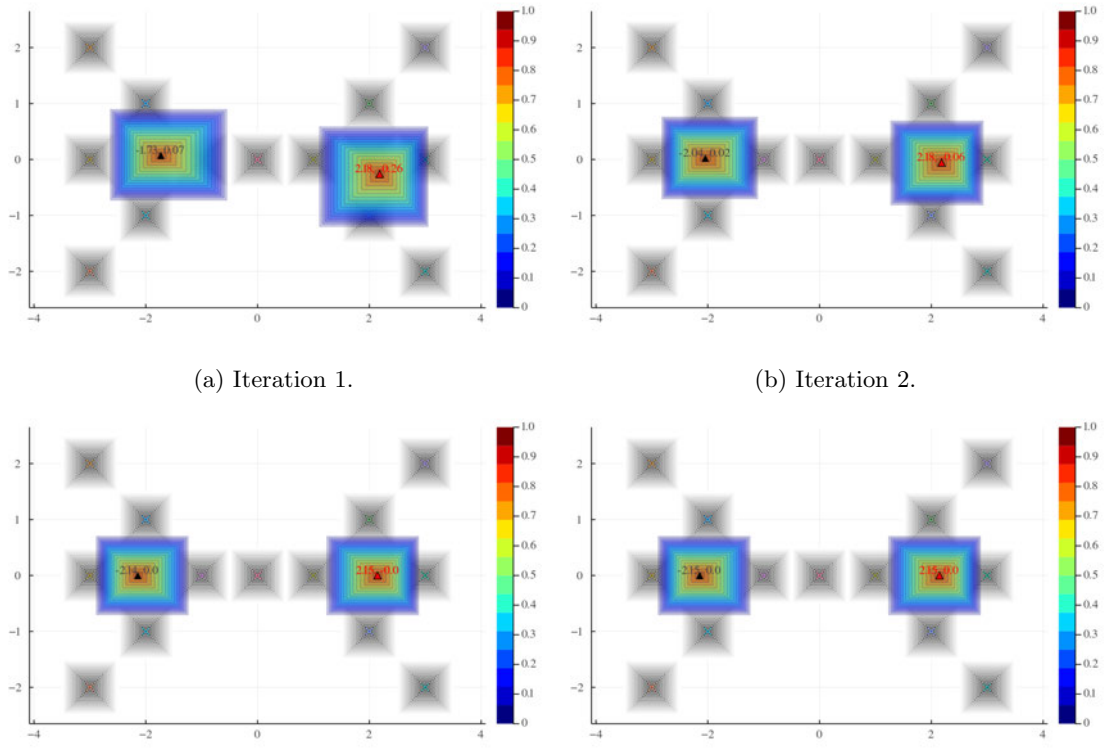


(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.35: The reflection dampened LFCM membership functions $u_{10,1}$ and $u_{10,2}$ with $m = 2$ and $\gamma = 0.5$ at different iterations. The black functions represent $u_{10,1}$, whereas the red functions represent $u_{10,2}$. They are both singleton fuzzy numbers at the beginning since the 10^{th} pattern is the initial cluster center, and they are not affected by the reflection dampening.

Apart from the memberships that are directly dampened using the reflection dampening, we also observe the indirect effect on the cluster centers. Here, the cluster centers, calculated from the

reflection dampened memberships, have smaller uncertainty compared to the original ones. This is the indirectly effect of applying such a dampening method to the memberships.



(a) Iteration 1.

(b) Iteration 2.

(c) Iteration 5.

(d) Iteration 10.

Figure 3.36: The resulting cluster centers (colored contours) from running the LFCM with $m = 2$ on the fuzzified butterfly dataset (black contours) at different iterations. The memberships used to compute these cluster centers are dampened with the reflection dampening approach.

Figure 3.37 is a plot of U-uncertainties of the first cluster center over iterations when the LPCM with $m = 2$ is executed with different dampening methods. $U(\vec{C}_1)$ obviously increases over time with no dampening. If one of the three dampening approaches is applied on the membership functions, the uncertainties are much less. Among the methods, the vertical cut with $\beta = 0.8$ has the strongest effect that shrinks the uncertainties, followed by the reflection dampening. Since the U-uncertainty is computed based on the nonspecificity of a fuzzy set, the linear dampening approach—that always makes the resulting membership function a triangle—is likely to have higher uncertainty than the other approaches (in this case $\gamma = 0.5$). As we know from the membership spread problem, one side of a membership function tends to be a long tail. While the vertical cut dampening and the reflection dampening remove the long tail side, the linear dampening does not cut the side but just dampens it to some degree, see also Figure 3.21.

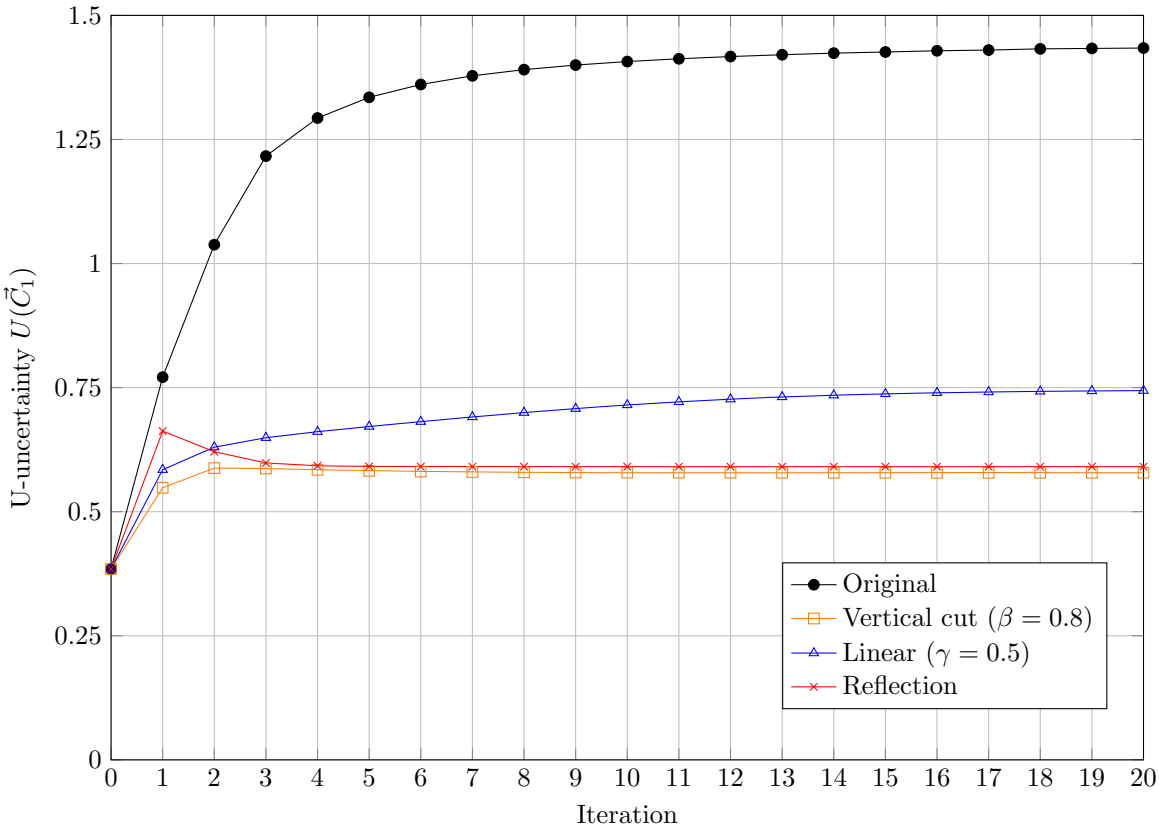


Figure 3.37: Plot of U-uncertainty values of the first cluster center $\vec{C}_1$ from the LFCM with $m = 2$ and different dampening methods on the fuzzified butterfly dataset.

To summarize, the vertical cut dampening is the simplest dampening to implement among the three methods. It removes the level cuts lower than β by setting them to be equal to ${}^{\beta}[u_{ji}]$. The linear dampening estimates the slopes around the peak and dampen the slopes to some degree. It can be employed, not only to mitigate the membership spread problem, but also reduce the computational complexity if we apply interpolation between α-cuts. The third dampening method, reflection dampening, is capable of preserving the original shape of the membership by reflecting the side (from the peak) with the lesser uncertainty and reflect it around the peak. It also does not need any parameter to set to dampen the memberships

So, we are able to maintain the uncertainty in the memberships as the membership functions can be dampened using the dampening approaches. They result in the thinner membership functions when compared to the original ones. In other words, the uncertainty from the characteristic of LFCM and LPCM is effectively suppressed, not only at a single iteration, but also over iterations. In addition, they also have indirect effects on the updated cluster centers. The uncertainty associated with the cluster centers can be shrunk, especially for the lower level cuts.

3.3 Linguistic Possibilistic C-Means

As we saw the memberships and cluster centers produced from the LFCM with different m values, and also with and without dampening methods, it would be interesting to see how the LFCM works if the dataset contains some noise. Therefore, we add an outlier to the fuzzified butterfly patterns as the 16^{th} pattern. We call this dataset the butterfly+outlier dataset. Figure 3.38 depicts how the dataset looks like after the outlier at $[0, 10]$ is added.

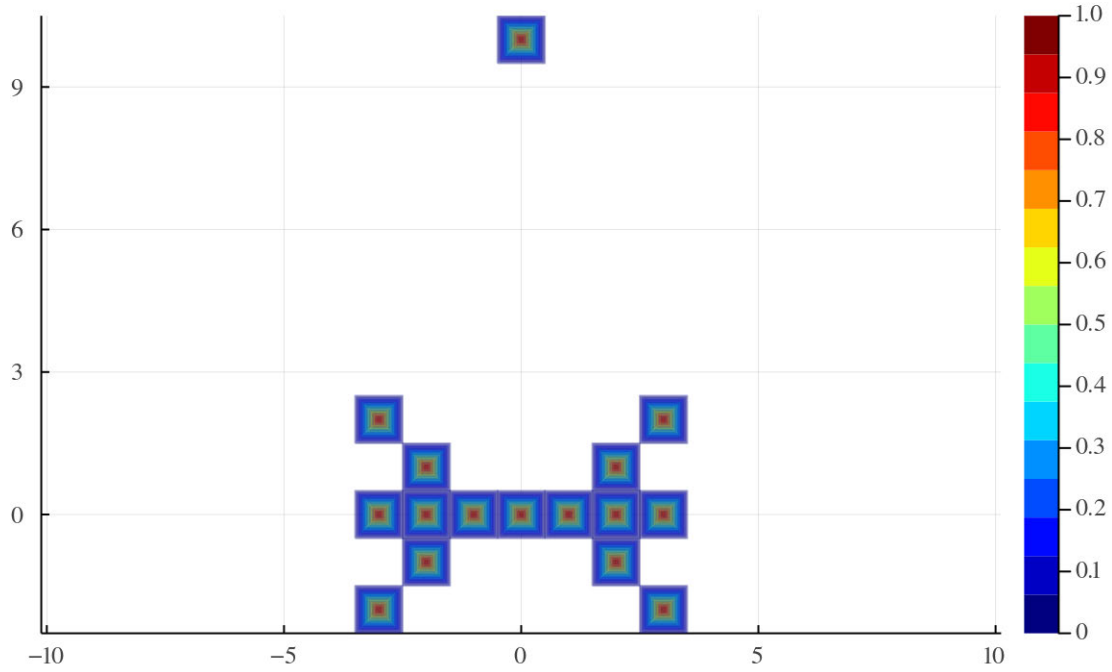
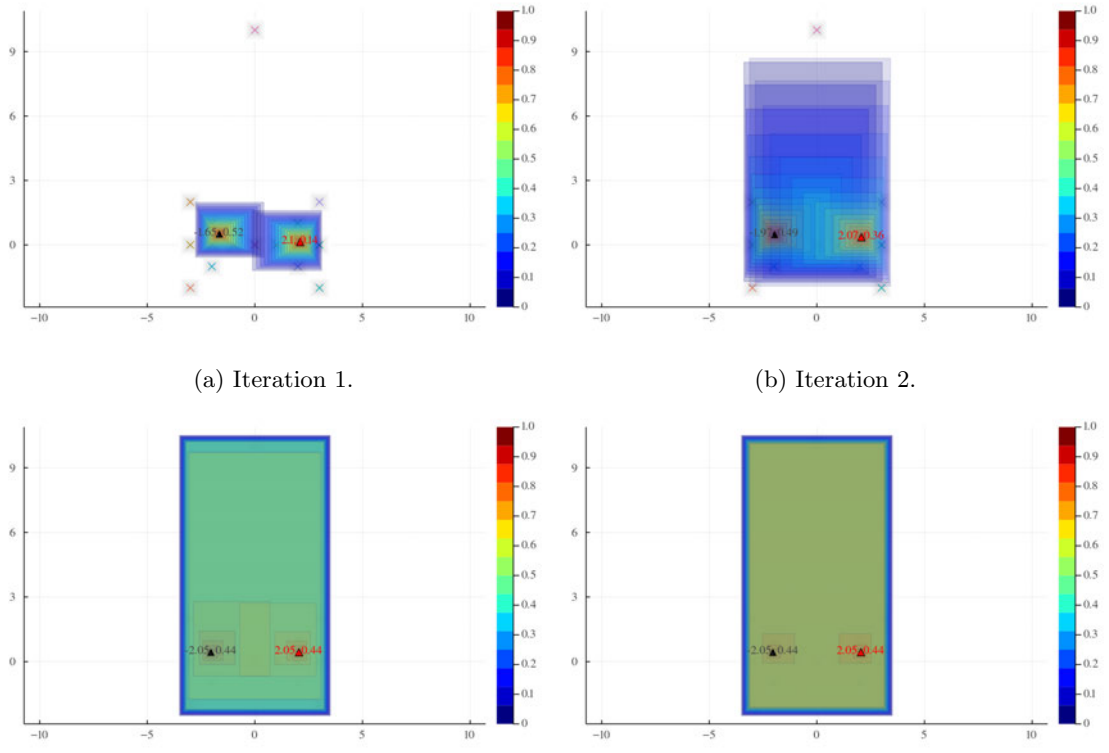


Figure 3.38: The top-down contour plot of the fuzzified butterfly+outlier dataset visualized in 2-dimensional space. There are 15+1 crisp points that are converted into 16 fuzzy vectors. Triangular fuzzy number is used in each dimension. The first 15 fuzzy vectors are indexed as shown Figure 3.1. The outlier is labeled as the 16^{th} fuzzy vector.

The LFCM with $m = 2$

Figure 3.39 provides the type-2 cluster centers at different iterations resulting from the LFCM with $m = 2$. We can see that, at iteration 2, the uncertainty grows very fast toward the outlier (along the y-axis). The peaks of the final cluster centers are $(-2.05, 0.44)$ and $(2.05, 0.44)$ which differ from those of the fuzzifier butterfly dataset without the outlier. The y locations are pulled up from 0.00 to 0.44 as they are influenced by the outlier, even it is further away. According to this experiment

that involves only one outlier, it demonstrates that the LFCM is susceptible to noise.



(a) Iteration 1.

(b) Iteration 2.

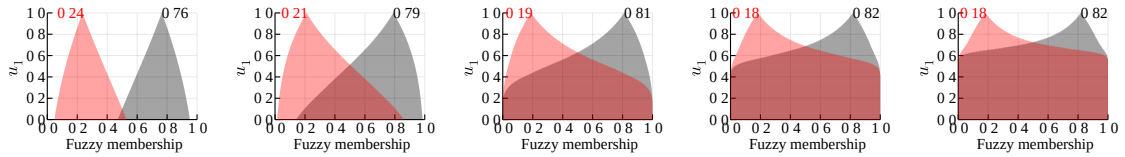
(c) Iteration 5.

(d) Iteration 10.

Figure 3.39: The resulting cluster centers (colored contours) from running the LFCM with $m = 2$ on the fuzzified butterfly+outlier dataset (black contours) at different iterations.

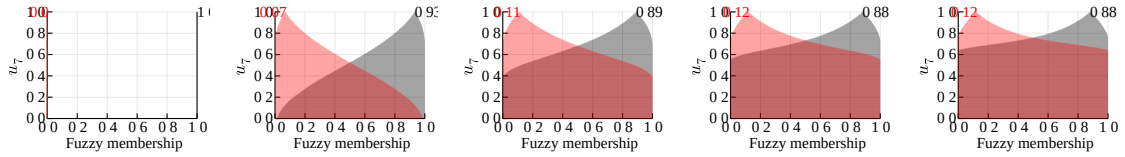
The LFCM memberships u_1, u_7, u_8 and u_{10} of the LFCM with $m = 2$ on the dataset are portrayed in Figures 3.40–3.43. Comparing them against that of the LFCM with $m = 2$ on the butterfly dataset does not give us much noticeable difference.

Even more importantly, the added outlier fuzzy vector in the butterfly+outlier dataset has an influence on the resulting cluster centers according to (2.44). Figure 3.44 shows the membership functions $u_{16,1}$ and $u_{16,2}$ from iteration 1 to iteration 5. Since the relative distances from the outlier $\vec{X}_{16}$ to the cluster centers are equal, then the memberships are about 0.5 in the end. It is the same thing happens with u_8, but they are different "about 0.5" fuzzy sets, see Figure 3.42e versus Figure 3.44e.



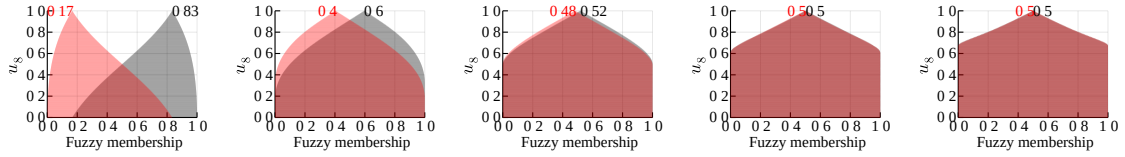
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.40: The membership functions $u_{1,1}$ and $u_{1,2}$ from running the LFCM with $m = 2$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{1,1}$, whereas the red functions represent $u_{1,2}$.



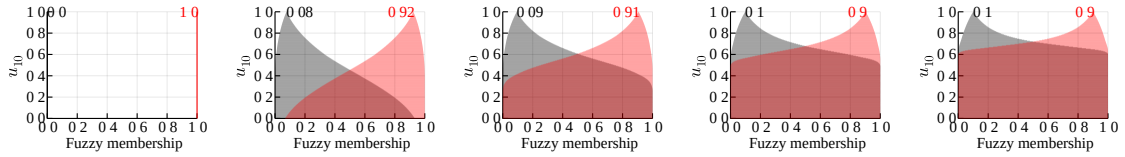
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.41: The membership functions $u_{7,1}$ and $u_{7,2}$ from running the LFCM with $m = 2$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{7,1}$, whereas the red functions represent $u_{7,2}$.



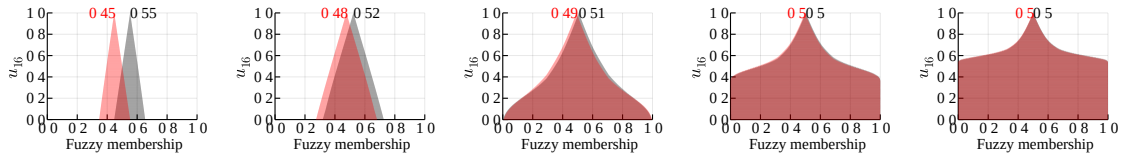
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.42: The membership functions $u_{8,1}$ and $u_{8,2}$ from running the LFCM with $m = 2$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{8,1}$, whereas the red functions represent $u_{8,2}$.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.43: The membership functions $u_{10,1}$ and $u_{10,2}$ from running the LFCM with $m = 2$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{10,1}$, whereas the red functions represent $u_{10,2}$.



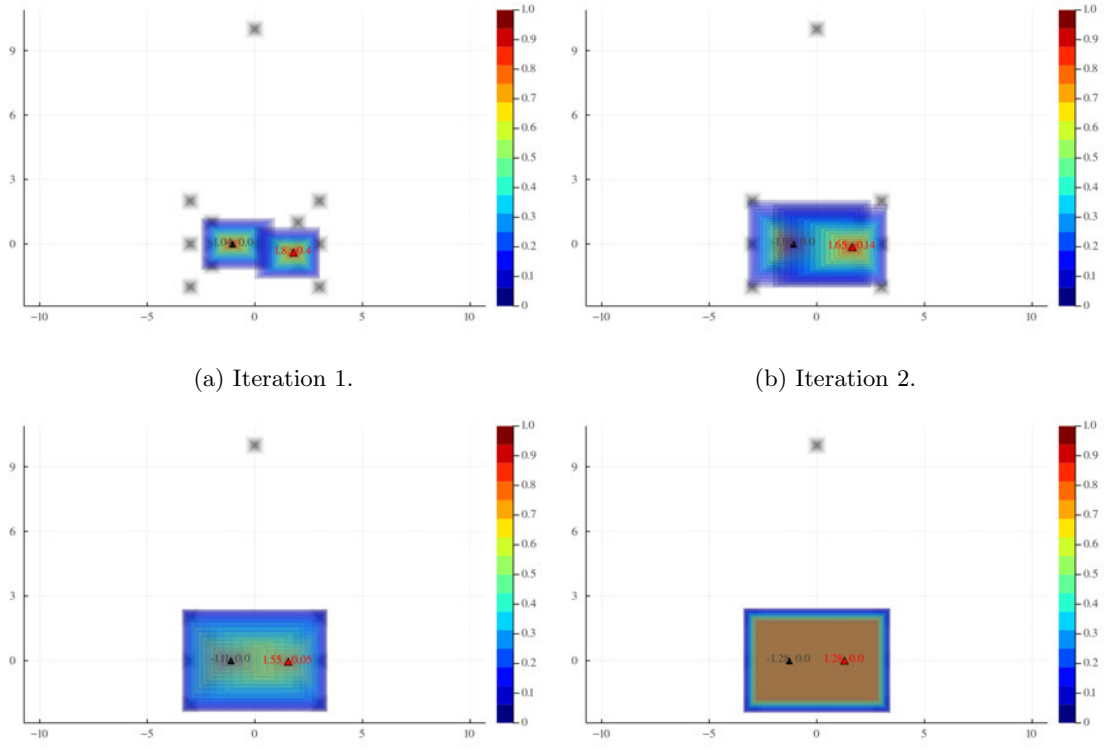
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 5.

Figure 3.44: The membership functions $u_{16,1}$ and $u_{16,2}$ from running the LFCM with $m = 2$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{16,1}$, whereas the red functions represent $u_{16,2}$.

The LPCM with $m = 2$ and $\eta_1 = \eta_2 = 4$

Now, here comes the LPCM. What if we run the LPCM on the butterfly+outlier dataset with the same setting? Since the LPCM requires not only the fuzzifier m, but also the η_i to be set, we need to set η_i for $i = 1, 2$. However, finding a good choice of η_i is another challenging problem one would encounter when using PCM or LPCM. But it is more difficult for the latter, so we will not cover this issue in this study. Since we know from the PCM algorithm that the values of η_i determine the squared distances $(d_{ji})^2$ at which the memberships cross 0.5 (see $(d_{ji})^2/\eta_i$ in (2.49)), we will use this knowledge along with the crisp butterfly patterns. The crisp cluster centers are expected to be at $(-2, 0)$ and $(2, 0)$, respectively. The patterns 1-7 must belong to the first cluster, whereas the patterns 9-15 must belong to another cluster. The bridge point $\vec{X}_8$ could be in either cluster. Hence, we compute the squared Euclidean distance between $\vec{X}_5$ and $\vec{X}_8$ (between $(-2, 0)$ and $(0, 0)$), which is $(-2 - 0)^2 + (0 - 0)^2 = 4$. to set η_1 and η_2. All in all, we set $\eta_1 = \eta_2 = 4$ for the sake of simplicity in this experiment without varying the parameters. From now on, the LPCM cluster algorithm will run from scratch with no extra initialization (e.g., running LFCM beforehand).

Let us run the LPCM with $m = 2$ (the same fuzzifier value as the previous experiment) and $\eta_1 = \eta_2 = 4$. Figure 3.45 displays the LPCM cluster centers for the butterfly+outlier dataset at different iterations. The cores of the final cluster centers finally become $\langle(-1.28, 0.00), (1.28, 0.00)\rangle$ after the convergence at iteration 28. We also see that it requires more iterations, as contrasted to the case of LFCM, to reach the convergence. The uncertainty builds up rapidly, except the 1-cuts that are degenerate intervals. Nonetheless, all the α-cuts of the cluster centers do not shift toward the outlier. This does affirm that the LPCM is not sensitive to noise.



(a) Iteration 1.

(b) Iteration 2.

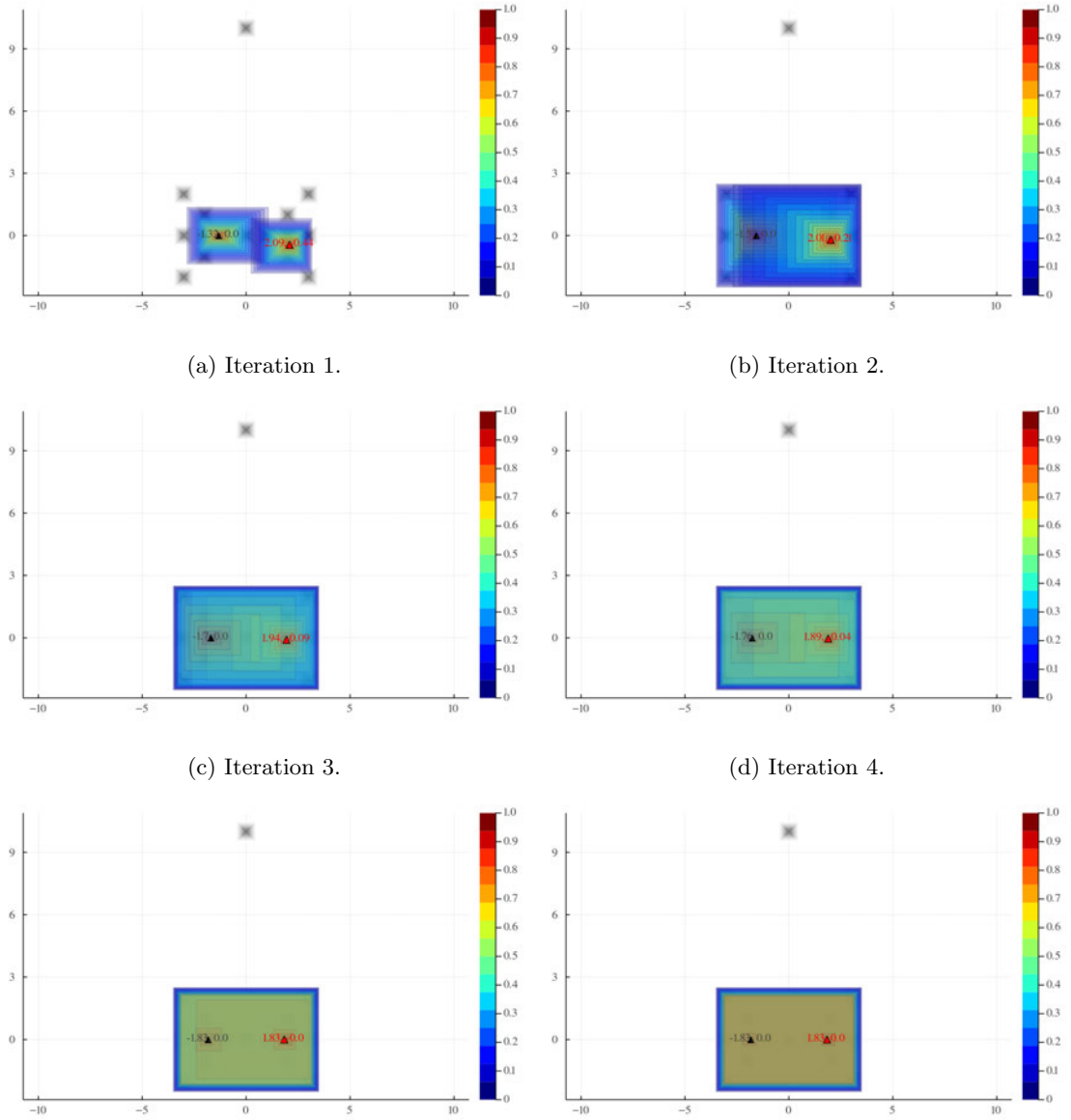
(c) Iteration 3.

(d) Iteration 28.

Figure 3.45: The resulting cluster centers (colored contours) from running the LPCM with $m = 2$ and $\eta_1 = \eta_2 = 4$ on the fuzzified butterfly+outlier dataset (black contours) at different iterations.

The LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$

As can be seen earlier, the cluster centers of the LPCM algorithm on the butterfly+outlier dataset with $m = 2$ and $\eta_1 = \eta_2 = 4$ are not the best. We expect their final cluster centers, especially the cores, to be about $\langle(-2.00, 0.00), (2.00, 0.00)\rangle$ which are close to that of the LFCM with $m = 2$. Now let's reduce the fuzzifier m to 1.25. Figure 3.46 portrays the cluster centers of the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ at various iterations. At iteration 2, the uncertainty grows swiftly for the lower α-cuts. The supports of the cluster centers span the patterns 1–15 at iteration 3. The cores of the cluster centers stop changing at iteration 10, and are located at $(-1.83, 0.00)$ and $(1.83, 0.00)$, respectively, as shown in Figure 3.46f. The convergence is reached at iteration 28 when all level cuts of the cluster centers is unchanging. This is much better than the LPCM with $m = 2$. Again, the α-cuts of the cluster centers do not shift toward the outlier. Hence, the LPCM is capable of handling noise.



(a) Iteration 1.

(b) Iteration 2.

(c) Iteration 3.

(d) Iteration 4.

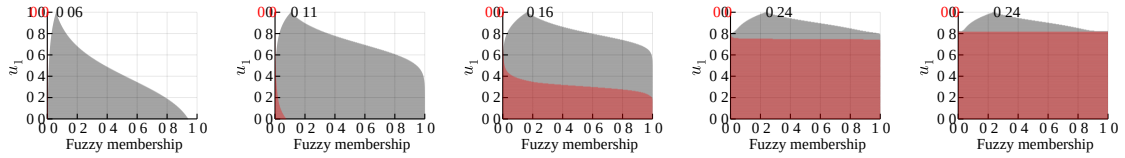
(e) Iteration 10.

(f) Iteration 28.

Figure 3.46: The resulting cluster centers (colored contours) from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the fuzzified butterfly+outlier dataset (black contours) at different iterations.

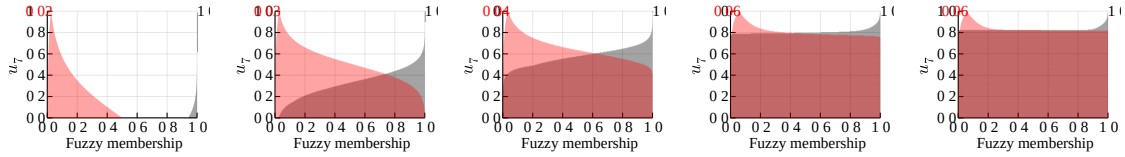
Like the LFCM algorithm, the memberships of LPCM also have the membership spread problem. Let's see the memberships (typicalities) from the LPCM with $m = 1.25$, $\eta_1 = \eta_2 = 4$ and no dampening method applied on the memberships. Figures 3.47–3.51 illustrate the memberships u_1, u_7, u_8, u_{10} and u_{16}. At iteration 1, The locations of $\vec{X}_1$ and $\vec{X}_7$ are closer to the first cluster center $\vec{C}_1$, then $u_{1,1} > u_{1,2}$, and $u_{7,1} > u_{7,2}$, considering the peaks. As the iterations progress, the uncertainty quickly builds up to the point where almost all α-cuts are uninformative for $\alpha \in [0, 0.8]$, i.e., they are

close to the interval $[0, 1]$. That is the case for all fuzzy vectors except the outlier. Nevertheless, the peaks of the memberships are still theoretically accurate. At iteration 28, the bridge point $\vec{X}_8$ has "about 0.67" memberships $u_{8,1}$ and $u_{8,2}$ which the cores come from $\frac{1}{1+(3.35/4)^{(1/(1.25-1))}} = 0.67$. We can see that $u_{8,1} = u_{8,2}$ because the relative distances from the bridge point to the cluster centers are equal (we have the perfect symmetry here).



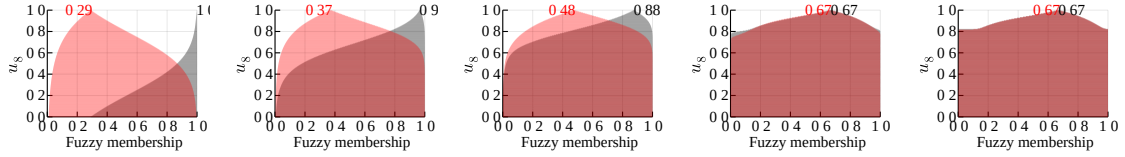
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 10. (e) Iteration 28.

Figure 3.47: The membership functions $u_{1,1}$ and $u_{1,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{1,1}$, whereas the red functions represent $u_{1,2}$.



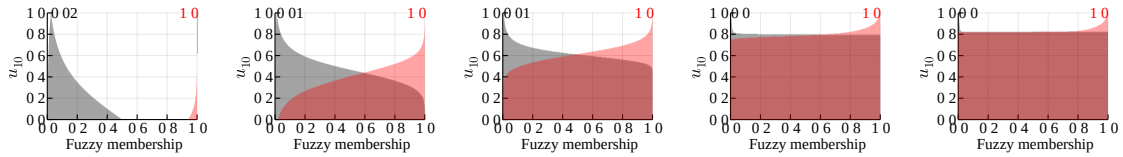
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 10. (e) Iteration 28.

Figure 3.48: The membership functions $u_{7,1}$ and $u_{7,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{7,1}$, whereas the red functions represent $u_{7,2}$.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 10. (e) Iteration 28.

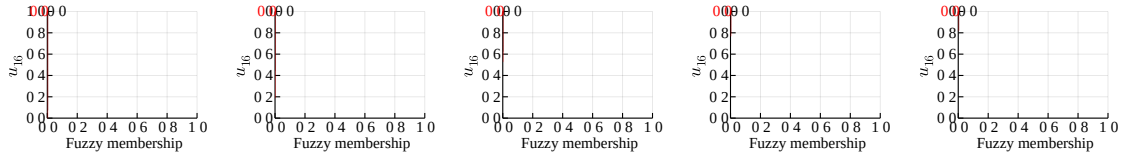
Figure 3.49: The membership functions $u_{8,1}$ and $u_{8,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{8,1}$, whereas the red functions represent $u_{8,2}$.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 10. (e) Iteration 28.

Figure 3.50: The membership functions $u_{10,1}$ and $u_{10,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{10,1}$, whereas the red functions represent $u_{10,2}$.

Figure 3.51 shows that the LPCM memberships $u_{16,1}$ and $u_{16,2}$ of the outlier $\vec{X}_{16}$ are fuzzy number "about 0", thin, and very close to singleton fuzzy number zero, at all iterations. The reason is that the outlier $\vec{X}_{16}$ are very far away from the two cluster centers $\vec{C}_1$ and $\vec{C}_2$.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 10. (e) Iteration 28.

Figure 3.51: The membership functions $u_{16,1}$ and $u_{16,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{16,1}$, whereas the red functions represent $u_{16,2}$.

Also, $\eta_i = 4$ for $i = 1, 2$ obviously have the influence on the LPCM memberships. Figure 3.52 displays the plot of the PCM typicalities when $m = 1.25$ and $\eta_i = 4$. If the squared distance is about 15, the typicality approaches zero. The core of the outlier $\vec{X}_{16}$ is $(0.00, 10.00)$. The core of the first cluster center at the convergence is $(-1.83, 0.00)$. The squared Euclidean distance between the two is $(d_{16,1})^2 = (0.00 - (-1.83))^2 + (10.00 - 0.00)^2 = 103.35$ which is very large. It is also the case for $(d_{16,2})^2$ because the dataset is symmetric. Hence, the typicalities $u_{16,1}$ and $u_{16,2}$ are very small and almost identical to singleton fuzzy number zero.

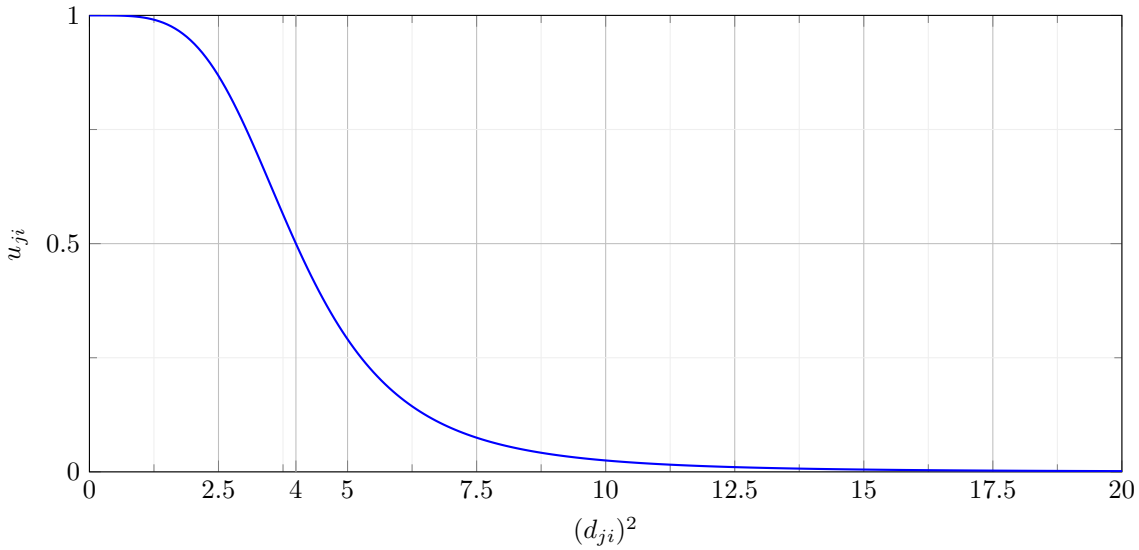


Figure 3.52: Plot of (2.49) when $\eta_i = 4$.

Now, what if we apply some dampening approaches to the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$. In this experiment, we choose the reflection dampening approach to solve the membership spread problem because it does not require any parameter and also preserve the original shapes of memberships to some extent.

Figure 3.53 shows the plot of U-uncertainties of the LPCM with $m = 1.25$ and different dampening methods. One could observe that the reflection dampening here results in the least uncertainties $U(\vec{C}_1)$ over iterations. However, the vertical cut with $\beta = 0.8$ in this case results in undesirable uncertainties. That is, after iteration 12, the memberships spread problem unexpectedly comes back, and shortly after that, the two cluster centers span the whole space again (now they are identical to the original ones). It also shows that the uncertainty associated with LPCM is more problematic than the uncertainty involved in LFCM. Then, we show the clustering results of the LPCM with $m = 1.25$ in combination with the reflection dampening next.

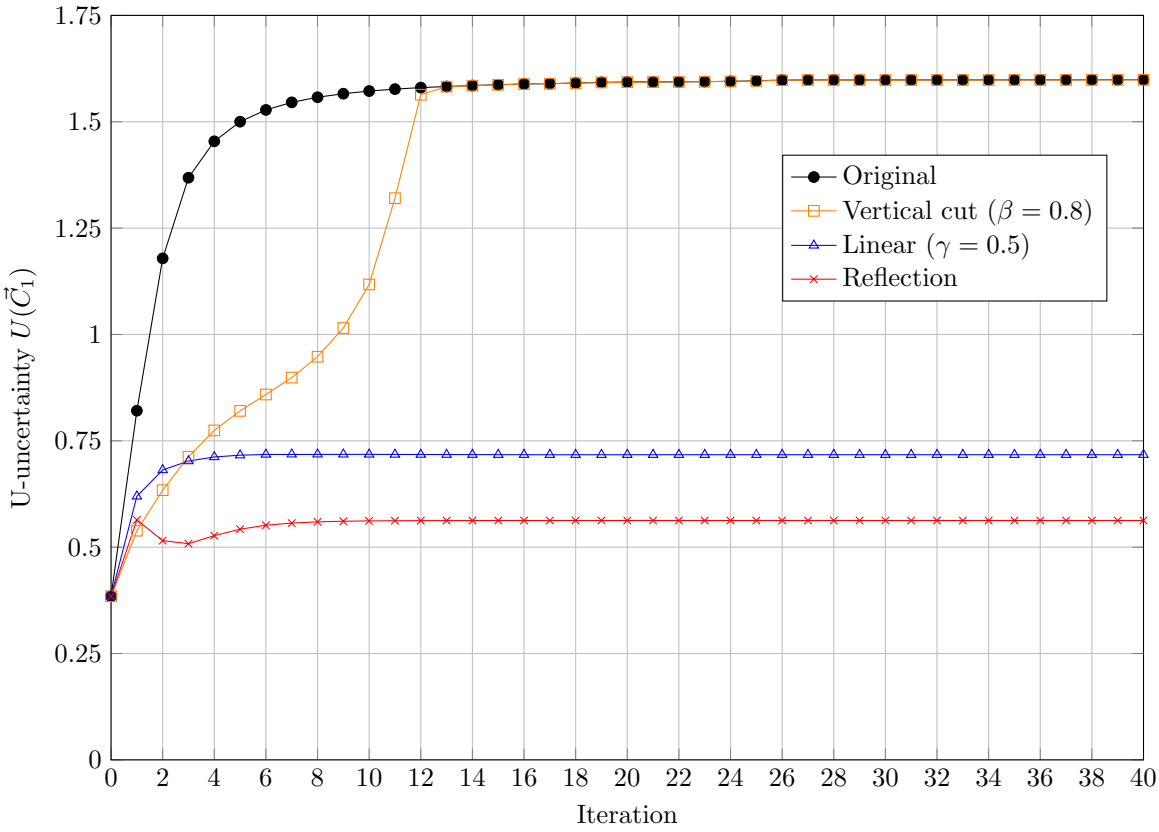
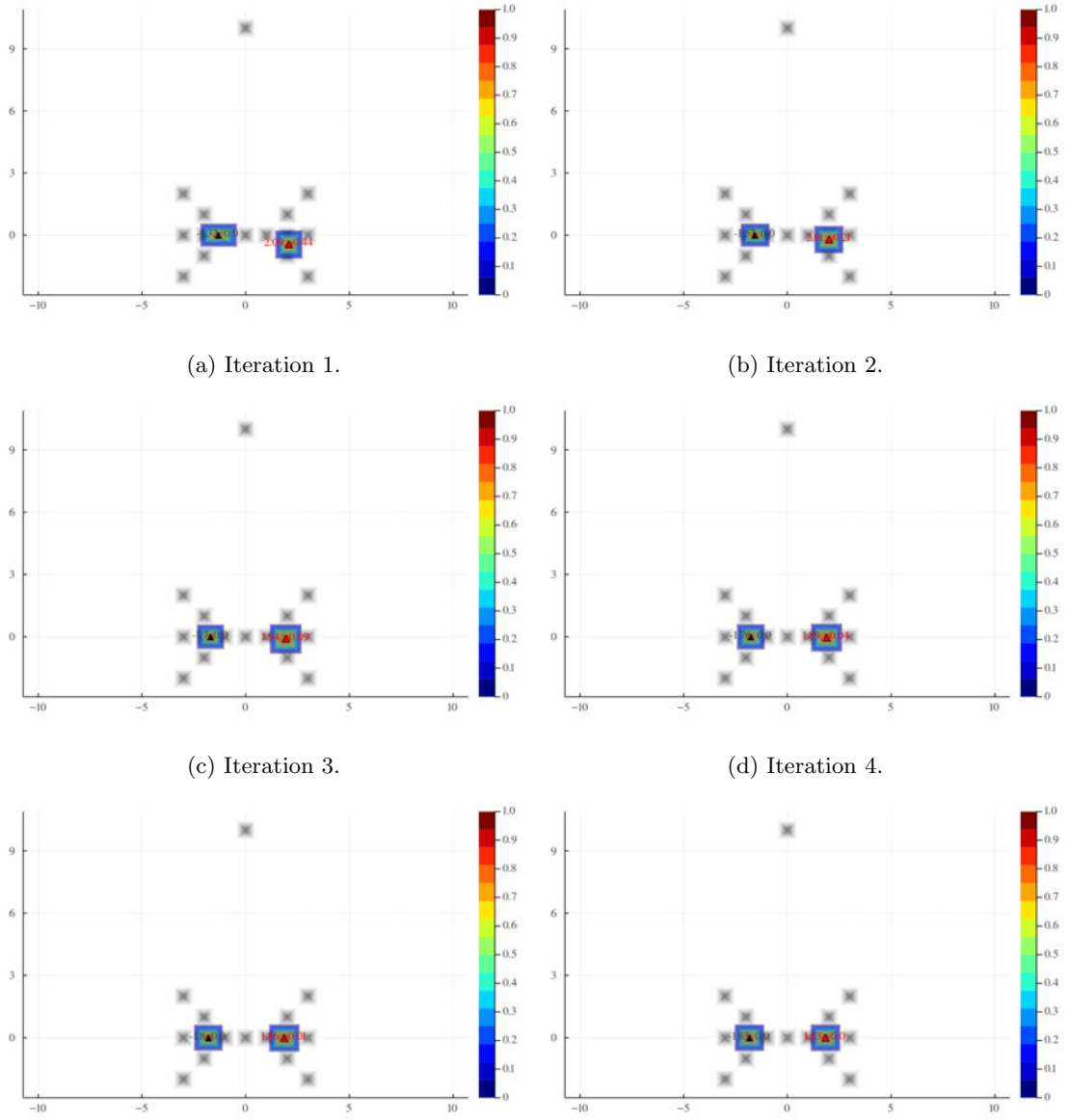


Figure 3.53: Plot of U-uncertainty values of the first cluster center $\vec{C}_1$ from the LPCM with $m = 1.25$ and different dampening methods.

The LPCM with $m = 1.25$, $\eta_1 = \eta_2 = 4$ and the reflection dampening

Figure 3.54 shows the clustering results of the LPCM with $m = 1.25$, $\eta_1 = \eta_2 = 4$ and the reflection dampening employed on the memberships. The good news here is that it reaches convergence much faster (10 iterations versus 28 iterations) considering all α-cuts of the cluster centers. Although the LPCM with $m = 1.25$ without dampening on the memberships no longer shifts the 1-cuts of the cluster centers after iteration 10, it needs 18 more iterations to actually converge if all α-cuts are taken into account. This is why the dampening methods are important as they can not only mitigate the membership spread problem directly, but also aid in convergence indirectly.



(a) Iteration 1.

(b) Iteration 2.

(c) Iteration 3.

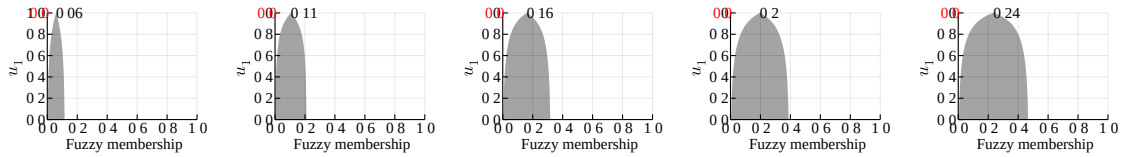
(d) Iteration 4.

(e) Iteration 5.

(f) Iteration 10.

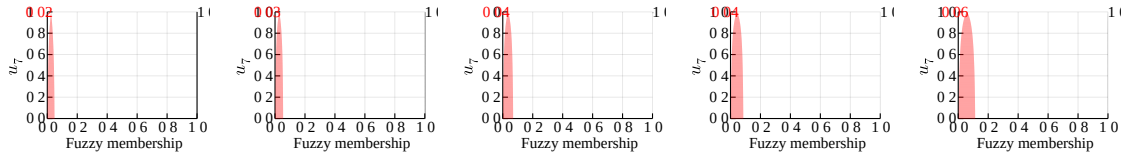
Figure 3.54: The resulting cluster centers (colored contours) from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the fuzzified butterfly+outlier dataset (black contours) at different iterations. The memberships used to compute these cluster centers are dampened with the reflection dampening approach.

Figures 3.55–3.59 shows the membership functions u_{ji} where $j \in \{1, 7, 8, 10, 16\}$ and $i \in \{1, 2\}$ of the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ with the reflection dampening. The cores of the original memberships u_7 and u_{10} (see Figure 3.48 and Figure 3.50) are either zero or unity. So, with reflection dampening, the resulting memberships are almost singleton fuzzy numbers, but they are not true singletons.



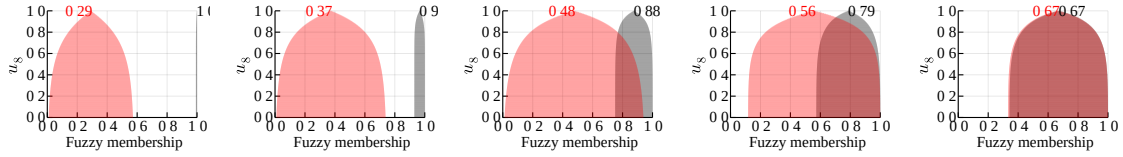
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 10.

Figure 3.55: The reflection dampened membership functions $u_{1,1}$ and $u_{1,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{1,1}$, whereas the red functions represent $u_{1,2}$.



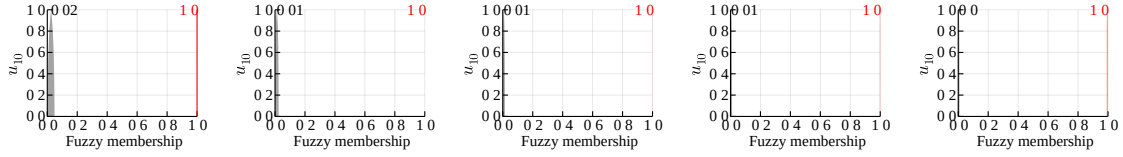
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 10.

Figure 3.56: The reflection dampened membership functions $u_{7,1}$ and $u_{7,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{7,1}$, whereas the red functions represent $u_{7,2}$.



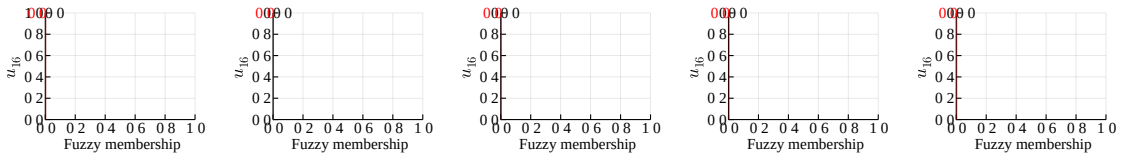
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 10.

Figure 3.57: The reflection dampened membership functions $u_{8,1}$ and $u_{8,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{8,1}$, whereas the red functions represent $u_{8,2}$.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 10.

Figure 3.58: The reflection dampened membership functions $u_{10,1}$ and $u_{10,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{10,1}$, whereas the red functions represent $u_{10,2}$.

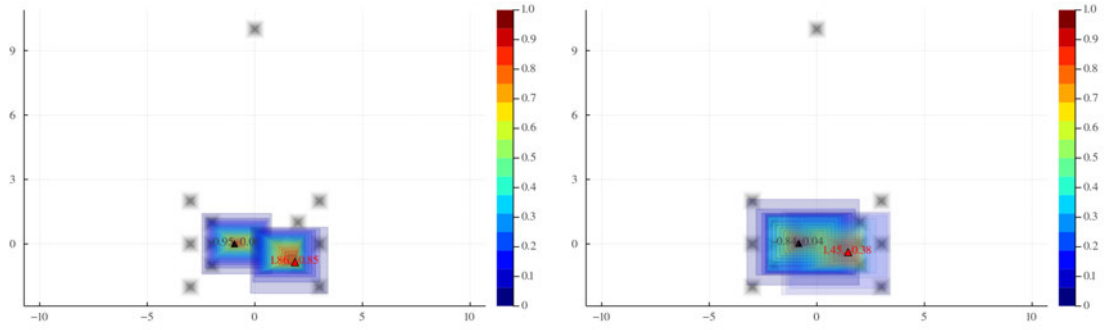


(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4. (e) Iteration 10.

Figure 3.59: The reflection dampened membership functions $u_{16,1}$ and $u_{16,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{16,1}$, whereas the red functions represent $u_{16,2}$. Note that the memberships in both cluster approach 0 quickly.

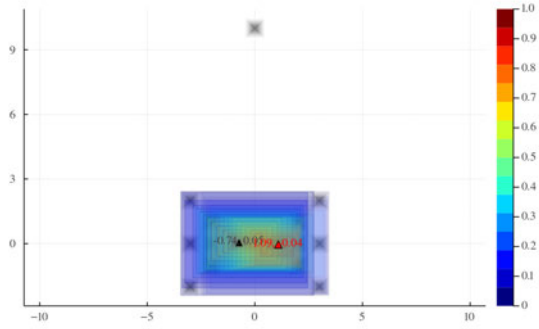
The LPCM with $m = 6$, $\eta_1 = \eta_2 = 4$

Figure 3.60 shows the results from running the LPCM clustering algorithm with the fuzzifier $m = 6$. Changing the fuzzifier m to 6 obviously affects the cluster prototypes. Iteration 3 is when we start to see the two cluster centers shift toward each together. As the iterations progress, they move closer to the bridge point, $\vec{X}_8$. Finally, we get coincident clusters at iteration 18 as displayed in Figure 3.60f. Even though, the 1-cuts of the cluster centers stop changing at iteration 18, the lower α-cuts still change.

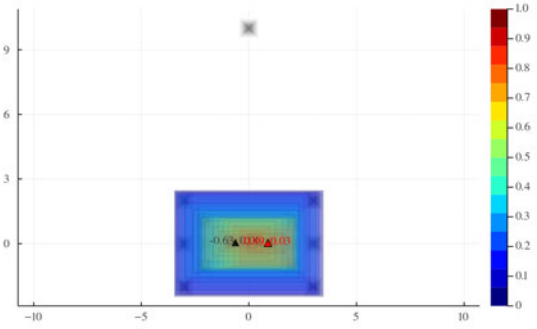


(a) Iteration 1.

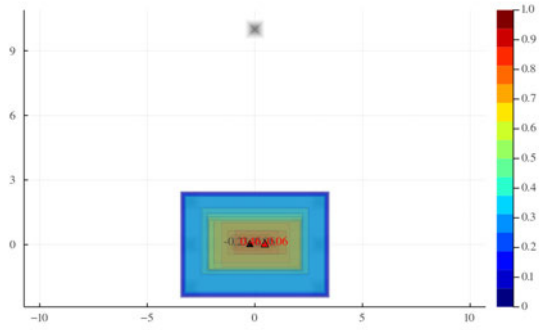
(b) Iteration 2.



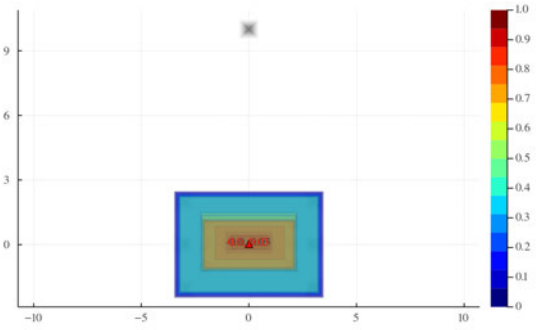
(c) Iteration 3.



(d) Iteration 4.



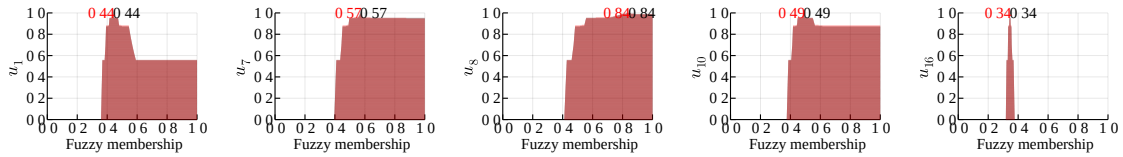
(e) Iteration 8.



(f) Iteration 18.

Figure 3.60: The resulting cluster centers (colored contours) from running the LPCM with $m = 6$ and $\eta_1 = \eta_2 = 4$ on the fuzzified butterfly+outlier dataset (black contours) at different iterations.

Figure 3.61 depicts the memberships u_{ji} where $j \in \{1, 7, 8, 10, 16\}$ and $i \in \{1, 2\}$ at iteration 40. Since the cluster centers form a coincident cluster, the LPCM memberships of each cluster in both clusters, $u_{j,1}$ and $u_{j,2}$, are identical for all $i = 1, 2, \ldots, 16$.

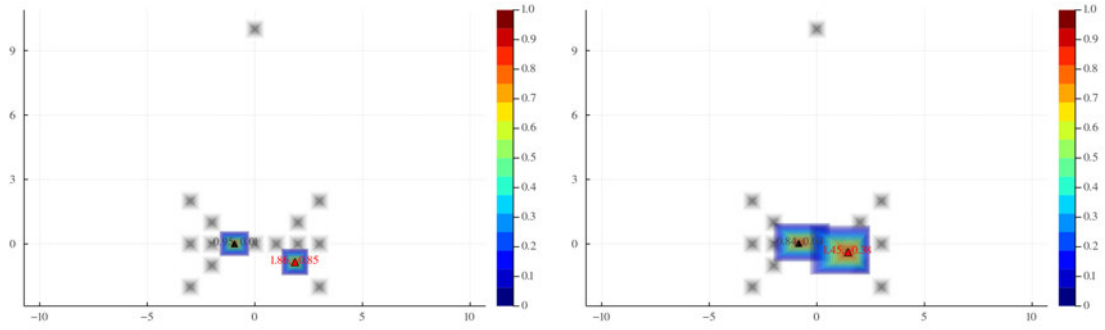


(a) $u_{1,i}$. (b) $u_{7,i}$. (c) $u_{8,i}$. (d) $u_{10,i}$. (e) $u_{16,i}$.

Figure 3.61: The membership functions from running the LPCM with $m = 6$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier dataset at iterations 40. The black functions represent $u_{j,1}$, whereas the red functions represent $u_{j,2}$ for $j \in \{1, 7, 8, 10, 16\}$.

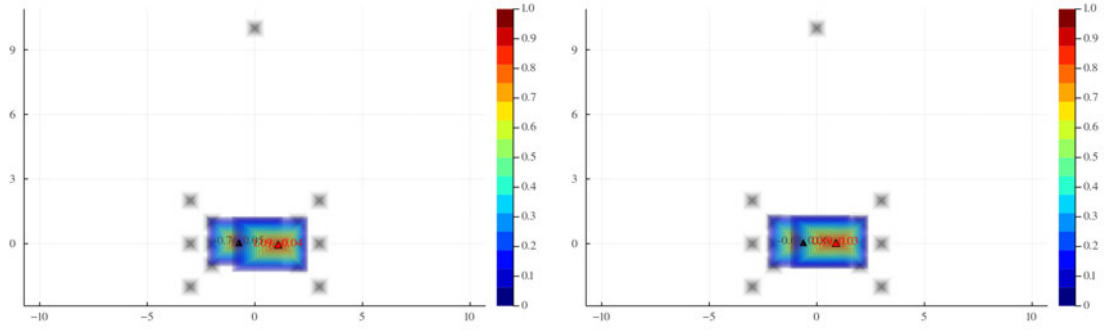
The LPCM with $m = 6$, $\eta_1 = \eta_2 = 4$ and the reflection dampening

Again, to shrink the uncertainty associated with the LPCM with $m = 6$ and $\eta_1 = \eta_2 = 4$, we apply reflection dampening on the membership functions in this experiment. Figure 3.62 shows the two cluster centers, $\vec{C}_1$ and $\vec{C}_2$, at different iterations. It is worthy of attention that the uncertainty of the coincident cluster centers shrinks after iteration 11, see Figure 3.62f, Figure 3.62g and Figure 3.62h. The coincident cluster centers obviously have less uncertainty at iteration 18 than that of iteration 11. Why does this happen? The answer lies in the membership functions which we will investigate next.



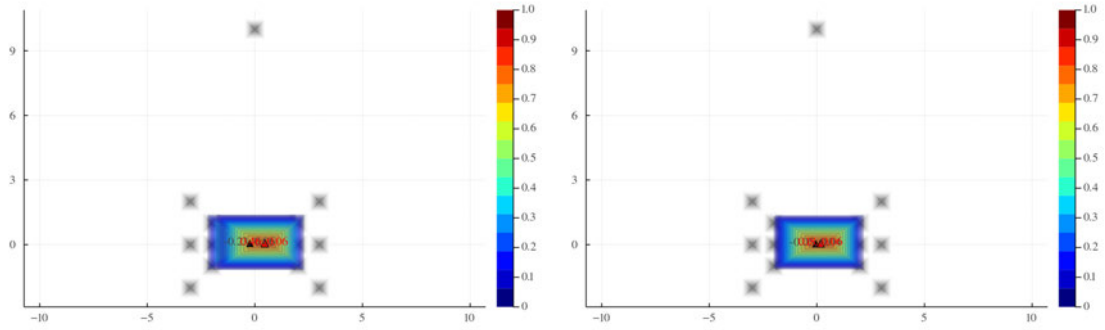
(a) Iteration 1.

(b) Iteration 2.



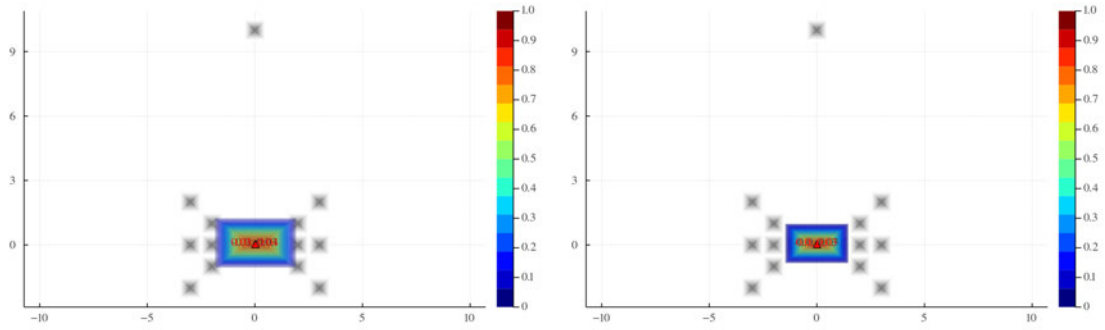
(c) Iteration 3.

(d) Iteration 4.



(e) Iteration 8.

(f) Iteration 11.

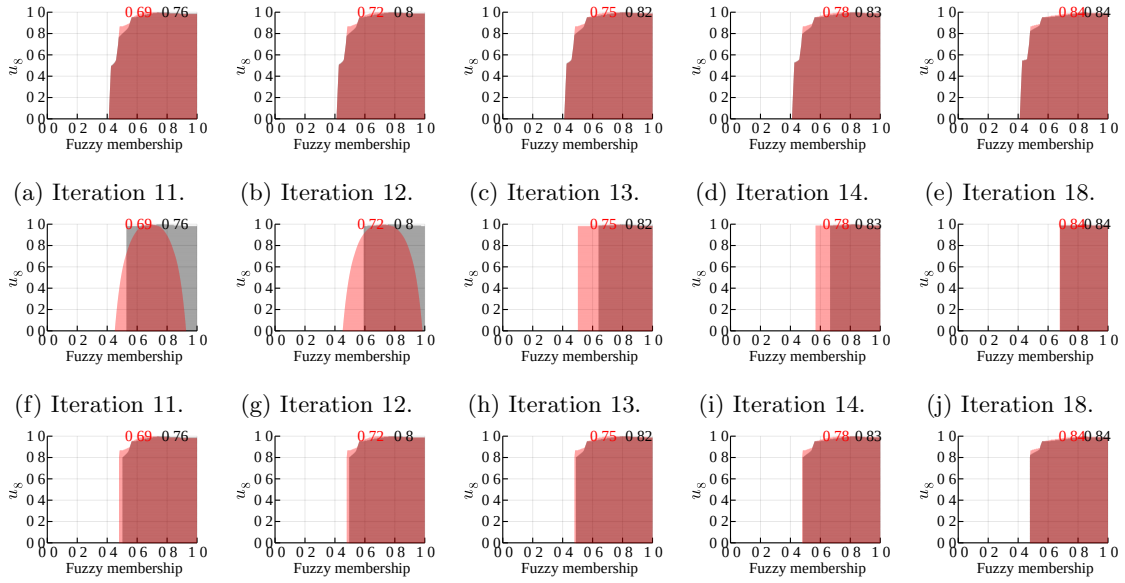


(g) Iteration 14.

(h) Iteration 18.

Figure 3.62: The resulting cluster centers (colored contours) from running the LPCM with $m = 6$ and $\eta_1 = \eta_2 = 4$ on the fuzzified butterfly+outlier dataset (black contours) at different iterations. The memberships used to compute these cluster centers are dampened with the reflection dampening approach.

Here, we focus on the LPCM membership functions $u_{8,1}$ and $u_{8,2}$. Figure 3.63 shows the comparison between the original memberships (the first row), the reflection dampened memberships (the second row), and the vertical cut dampened memberships (the third row), at iterations 11–14 and 18. The memberships $u_{8,1}$ and $u_{8,2}$ shift toward "about 1" (it is easier to look at only the 1-cuts) as the coincident cluster centers move to $(0.00, 0.00)$. Reflection dampening takes the thinner side of a membership and reflects across the peak. As the memberships move toward "about 1", the right sides are thinner. Thus, the consequence of the reflection dampening here is shrinking the uncertainty. This does not occur if we employ the vertical cut dampening.



(a) Iteration 11. (b) Iteration 12. (c) Iteration 13. (d) Iteration 14. (e) Iteration 18.

(f) Iteration 11. (g) Iteration 12. (h) Iteration 13. (i) Iteration 14. (j) Iteration 18.

(k) Iteration 11. (l) Iteration 12. (m) Iteration 13. (n) Iteration 14. (o) Iteration 18.

Figure 3.63: The LPCM membership functions $u_{8,1}$ and $u_{8,2}$ from running the LPCM with $m = 6$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier dataset at different iterations. The black functions represent $u_{1,1}$, whereas the red functions represent $u_{1,2}$. The first row, (a) to (e), shows the original memberships. The second row, (f) to (j), shows the reflection dampened memberships. The third row, (k) to (o) shows the vertical cut dampened memberships with $\beta = 0.8$.

Figure 3.64 shows the U-uncertainties of the first cluster center from the LPCM with various fuzzifier m values. Unlike the LFCM, shown in Figure 3.17, the uncertainty is not directly proportional to the value of m.

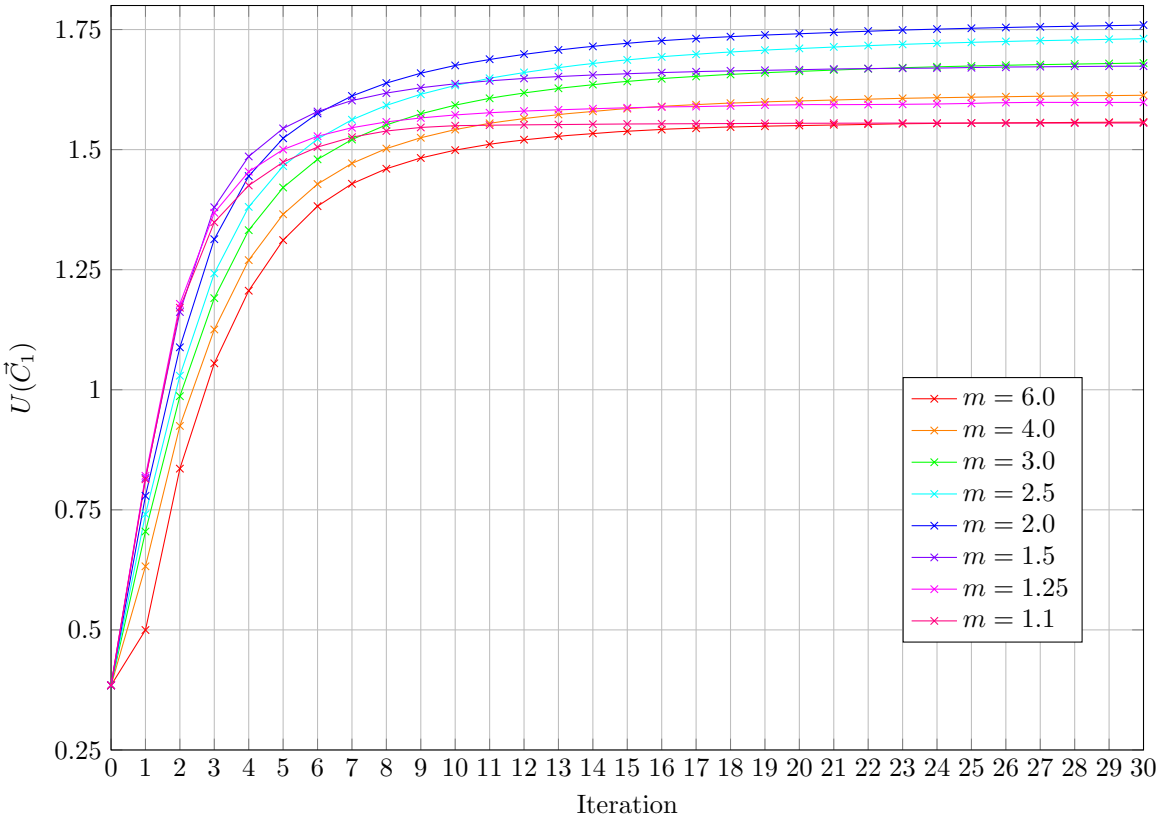


Figure 3.64: Plot of U-uncertainty values of the first cluster center, $U(\vec{C}_1)$, from the LPCM with different m values.

3.4 Data Fuzziness

Next, let us increase the fuzziness of each fuzzy vector in the butterfly+outlier dataset. It is done by increasing the width of all fuzzy numbers from 0.5 to 2.0. The butterfly+outlier is now altered to be a new dataset named butterfly+outlier2 with the widths of triangular fuzzy numbers being 2.0, or $w = 2.0$, as depicted in Figure 3.65. Earlier, each fuzzy vector does not overlap the others. Now, using $w = 2.0$, the input fuzzy vectors overlap nearby fuzzy vectors, i.e., at least the half of a fuzzy vector covers another fuzzy vector, except the outlier.

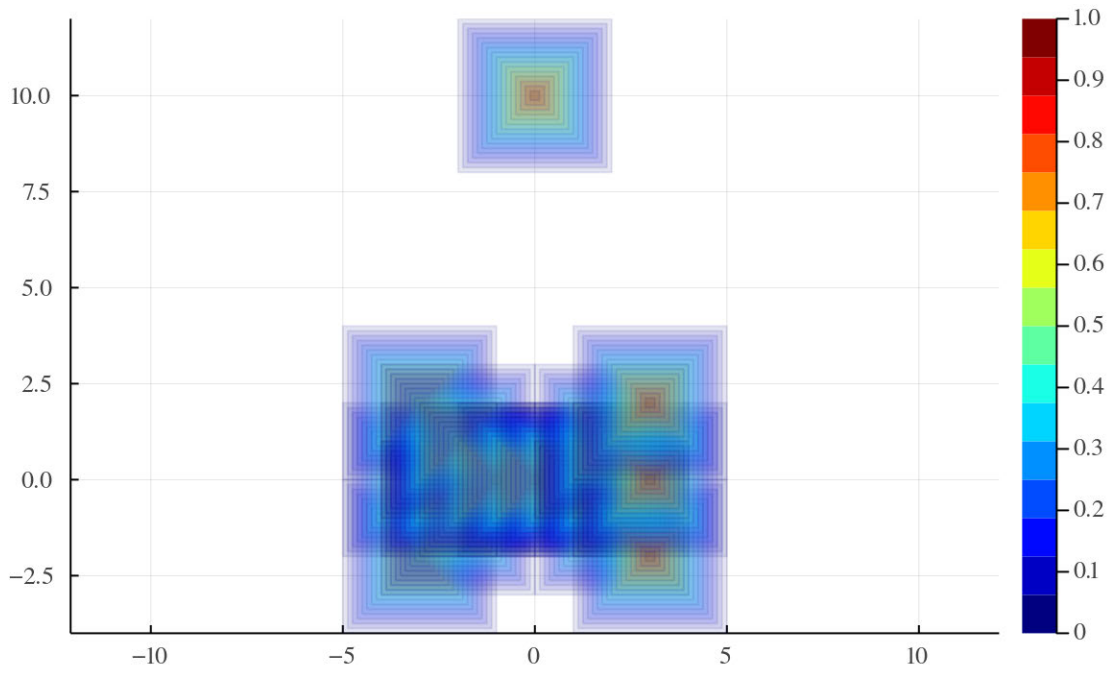
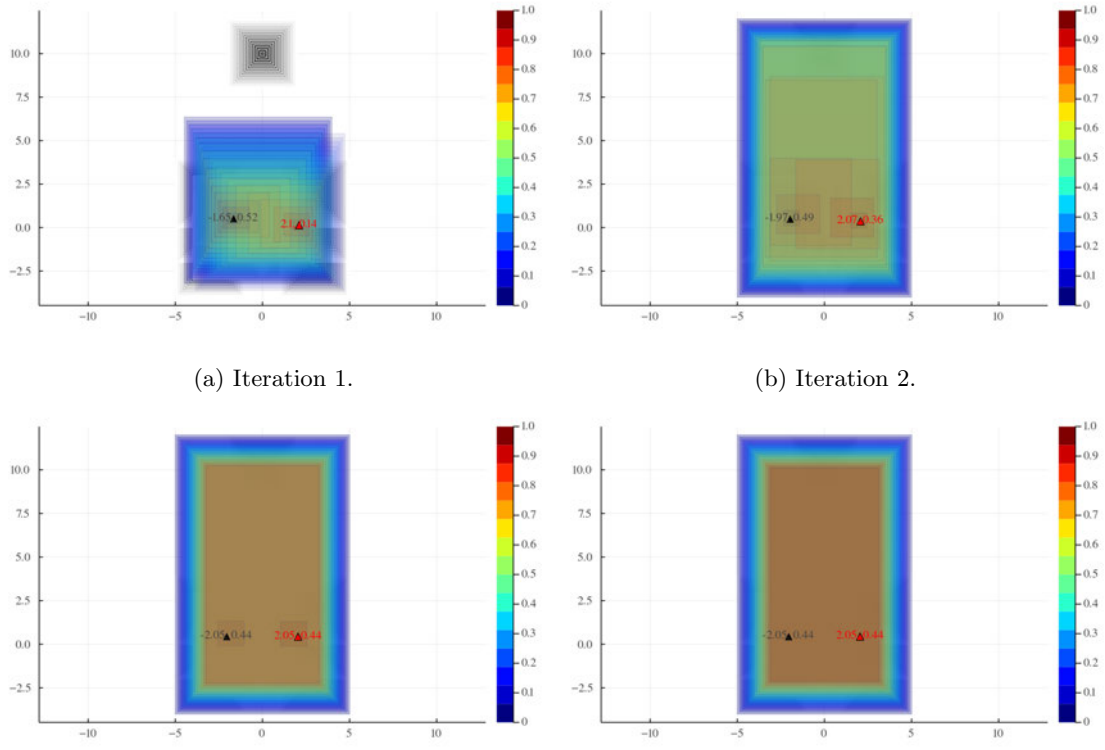


Figure 3.65: The butterfly+outlier2 dataset fuzzified using triangular fuzzy numbers with the width of 2.

The LFCM with $m = 2$

The LFCM with $m = 2$ is run on the butterfly+outlier2 dataset ($w = 2.0$). The resulting fuzzy vectors of cluster centers at different iterations are portrayed in Figure 3.66. We can compare them with the cluster centers from the butterfly+outlier dataset ($w = 0.5$) in Figure 3.39. As the fuzziness of the input fuzzy vectors are larger, the resulting cluster centers contains more uncertainty. Since we use triangular fuzzy numbers, the 1-cuts of the cluster centers are still exactly the same as when we use $w = 0.5$. That is the final cluster centers are pulled up toward the outlier, thus their cores end up at $(-2.05, 0.44)$ and $(2.05, 0.44)$, respectively, because the memberships depend on the relative distances to all input vectors. represented by the lower α-cuts explodes quickly.



(a) Iteration 1.

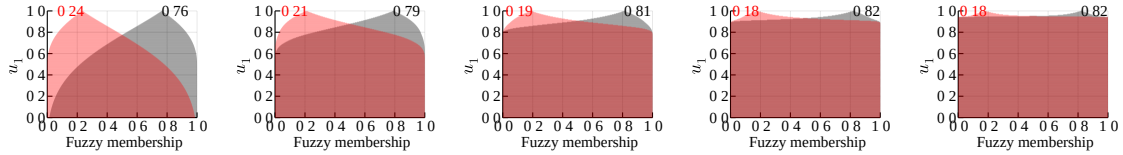
(b) Iteration 2.

(c) Iteration 5.

(d) Iteration 10.

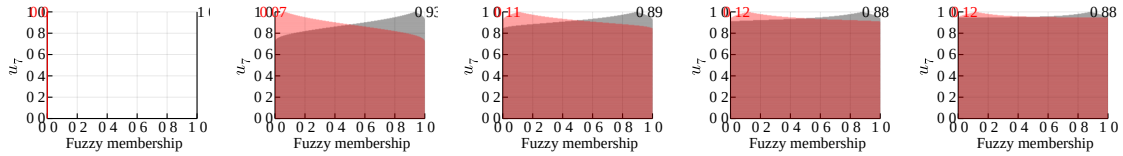
Figure 3.66: The resulting cluster centers (colored contours) from running the LFCM with $m = 2$ on the fuzzified butterfly+outlier2 dataset (black contours) at different iterations.

Now let us consider the memberships, u_1, u_7, u_8, u_{10} and u_{16} shown in Figure 3.67, Figure 3.68, Figure 3.69, Figure 3.70 and Figure 3.71, respectively. They can be compared against that of the LFCM with $m = 2$ on the butterfly+outlier dataset (that contains less uncertainty). At iteration 1, the memberships, except u_7 and u_{10} that are the initial cluster prototypes, have very large uncertainty. The α-cuts of $u_{8,1}$ and $u_{8,2}$ where $\alpha \in [0, 0.6]$ are obviously uninformative. The 1-cuts of the memberships are identical to that of the LFCM with $m = 2$ on the butterfly+outlier due to the fact that we use triangular fuzzy numbers to fuzzify the datasets. As we discussed, even at the first iteration, the uncertainty grows quickly. Hence, there is no doubt that the uncertainty at iteration 10 is very large. For example, the α-cuts of $u_{16,1}$ and $u_{16,2}$ where $\alpha \in [0.0, 0.9]$ are totally uninformative as they are $[0, 1]$. Only the cores and some lower level cuts are what we expect (based on the results from the regular FCM). This is where the dampening approaches can be employed to solve the uncertainty that are out of control.



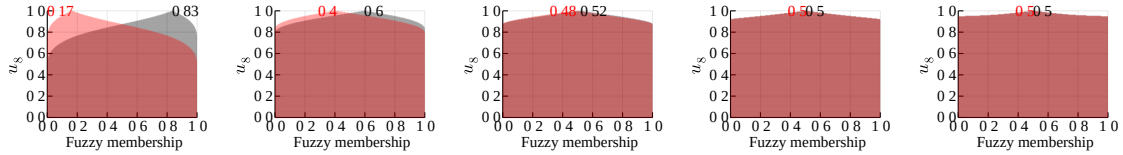
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 5. (e) Iteration 10.

Figure 3.67: The membership functions $u_{1,1}$ and $u_{1,2}$ from running the LFCM with $m = 2$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{1,1}$, whereas the red functions represent $u_{1,2}$.



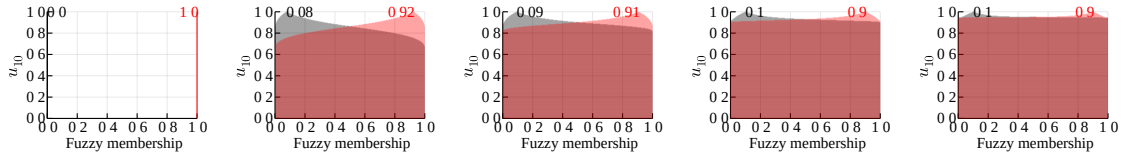
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 5. (e) Iteration 10.

Figure 3.68: The membership functions $u_{7,1}$ and $u_{7,2}$ from running the LFCM with $m = 2$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{7,1}$, whereas the red functions represent $u_{7,2}$.



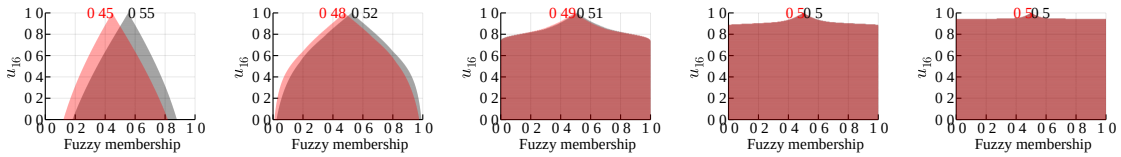
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 5. (e) Iteration 10.

Figure 3.69: The membership functions $u_{8,1}$ and $u_{8,2}$ from running the LFCM with $m = 2$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{8,1}$, whereas the red functions represent $u_{8,2}$.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 5. (e) Iteration 10.

Figure 3.70: The membership functions $u_{10,1}$ and $u_{10,2}$ from running the LFCM with $m = 2$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{10,1}$, whereas the red functions represent $u_{10,2}$.



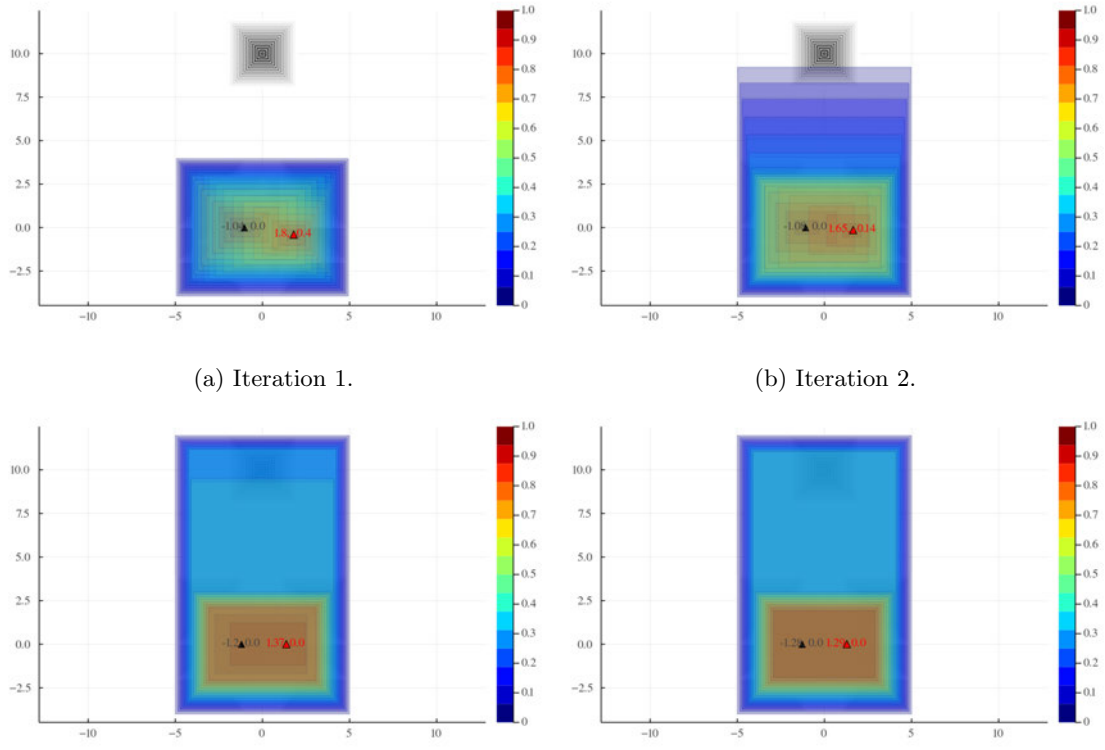
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 5. (e) Iteration 10.

Figure 3.71: The membership functions $u_{16,1}$ and $u_{16,2}$ from running the LFCM with $m = 2$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{16,1}$, whereas the red functions represent $u_{16,2}$.

The LPCM with $m = 2$ and $\eta_1 = \eta_2 = 4$

What about running the LPCM algorithm on the butterfly+outlier2 dataset? Here we choose $\eta_1 = \eta_2 = 4$ and set $m = 2$. Figure 3.72 show the clustering results at various iterations. We can obviously see the lower level cuts of the cluster centers shift toward the outlier starting from iteration 2. At iteration 5, they span the whole space including the outlier fuzzy vector. Since the input fuzzy vectors are fuzzier (as the width of each fuzzy number is large), the 1-cuts of the final cluster centers are $(-1.28, 0.00)$ and $(1.29, 0.00)$, respectively, at iteration 20. Actually, we expect the cores of the final cluster centers to be somewhere around $(-2.00, 0.00)$ and $(2.00, 0.00)$, respectively. Hence, this is likely that we will get a coincident cluster from running the LPCM on the butterfly+outlier2 dataset which contains high uncertainty.

Also, the lower level cuts of the cluster centers span the whole space including the outlier. Although the LPCM clustering algorithm is capable of dealing with noise, but in this case, the input fuzzy vectors are too uncertain to the point that some lower α-cuts are uncertain toward the noise.



(a) Iteration 1.

(b) Iteration 2.

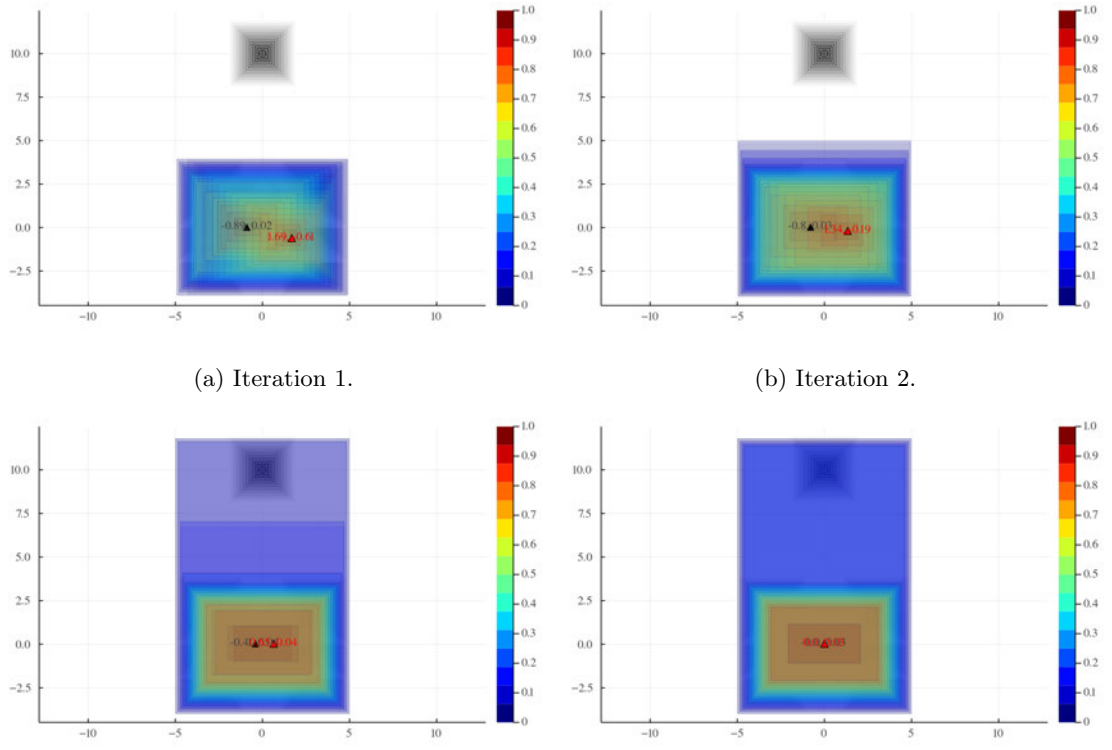
(c) Iteration 7.

(d) Iteration 20.

Figure 3.72: The resulting cluster centers (colored contours) from running the LPCM with $m = 2$ and $\eta_1 = \eta_2 = 4$ on the fuzzified butterfly+outlier2 dataset (black contours) at different iterations.

The LPCM with $m = 4$ and $\eta_1 = \eta_2 = 4$

So, to confirm that a coincident cluster can appear, we set the fuzzifier m to be equal to 4. Figure 3.73 depicts the clustering results at different iterations when LPCM is run with $m = 4$. The LPCM converges at iteration 31. If we consider only the cores of the final cluster centers, we can see that we can actually get a coincident cluster from LPCM. The butterfly+outlier2 dataset are more uncertain than the butterfly+outlier dataset. Due to the high uncertainty in the input fuzzy vectors, the consequence here is that the LPCM treat the patterns 1-15 as one big group. Thus, it yields the coincident cluster.



(a) Iteration 1.

(b) Iteration 2.

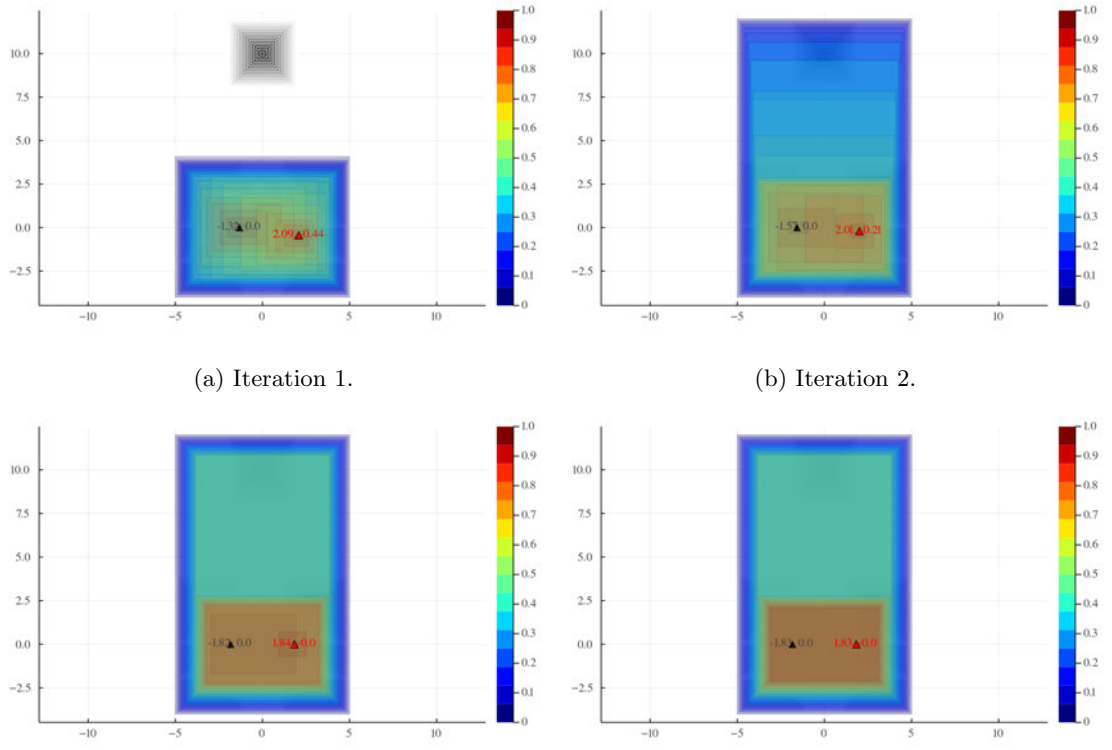
(c) Iteration 7.

(d) Iteration 31.

Figure 3.73: The resulting cluster centers (colored contours) from running the LPCM with $m = 4$ and $\eta_1 = \eta_2 = 4$ on the fuzzified butterfly+outlier2 dataset (black contours) at different iterations.

The LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$

We now reduce the fuzzifier m to 1.25 and expect not to see a coincident cluster at the end. Figure 3.74 illustrates the two cluster centers and the input fuzzy vectors at various iterations. As we expected, $m = 1.25$ in this case does not produce the coincident cluster which is what we please for this dataset that contains very high uncertainty. The cores of the final cluster centers are $(-1.83, 0.00)$ and $(1.83, 0.00)$, respectively, at iteration 13. Again, although the 1-cuts produce what we would expect from the LPCM on this dataset, the lower α-cuts of $\vec{C}_1$ and $\vec{C}_2$ still span the whole space due to the high uncertainty of the input fuzzy vectors.

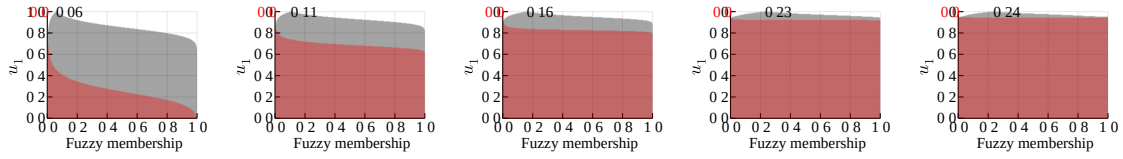


(a) Iteration 1. (b) Iteration 2.

(c) Iteration 7. (d) Iteration 13.

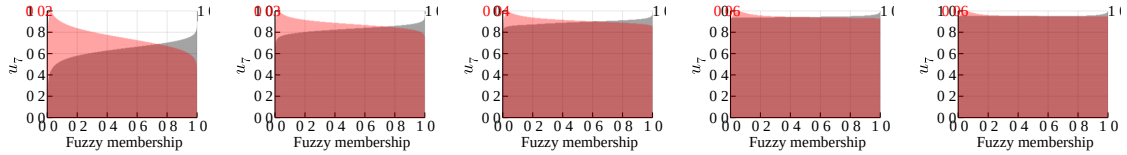
Figure 3.74: The resulting cluster centers (colored contours) from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the fuzzified butterfly+outlier2 dataset (black contours) at different iterations.

The LPCM with $m = 1.25$ in this case also produce very uncertain memberships for most of the α-cuts similar to that of the LPCM with $m = 2$. Here only the level cuts near the 1-cuts are informative as we expected. Since we have the dampening approaches, what would happen if we apply one of them on this setting.



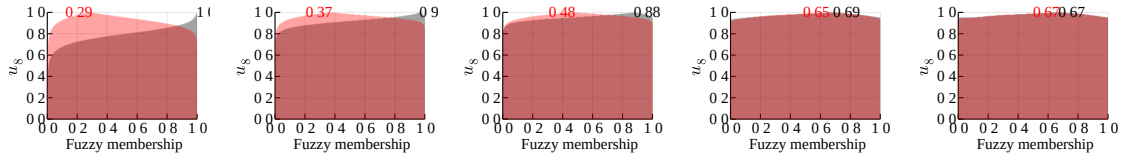
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 7. (e) Iteration 13.

Figure 3.75: The membership functions $u_{1,1}$ and $u_{1,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{1,1}$, whereas the red functions represent $u_{1,2}$.



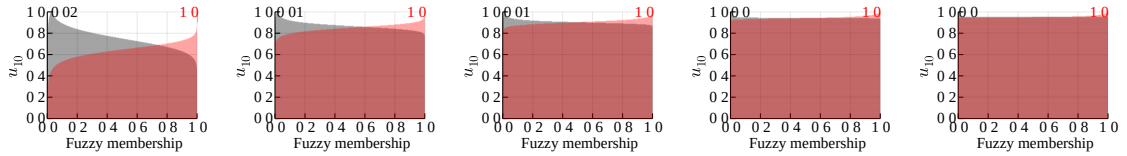
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 7. (e) Iteration 13.

Figure 3.76: The membership functions $u_{7,1}$ and $u_{7,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{7,1}$, whereas the red functions represent $u_{7,2}$.



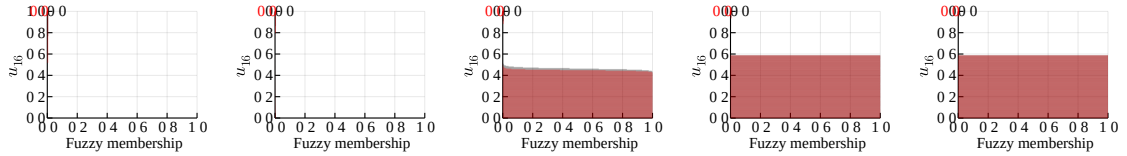
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 7. (e) Iteration 13.

Figure 3.77: The membership functions $u_{8,1}$ and $u_{8,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{8,1}$, whereas the red functions represent $u_{8,2}$.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 7. (e) Iteration 13.

Figure 3.78: The membership functions $u_{10,1}$ and $u_{10,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{10,1}$, whereas the red functions represent $u_{10,2}$.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 7. (e) Iteration 13.

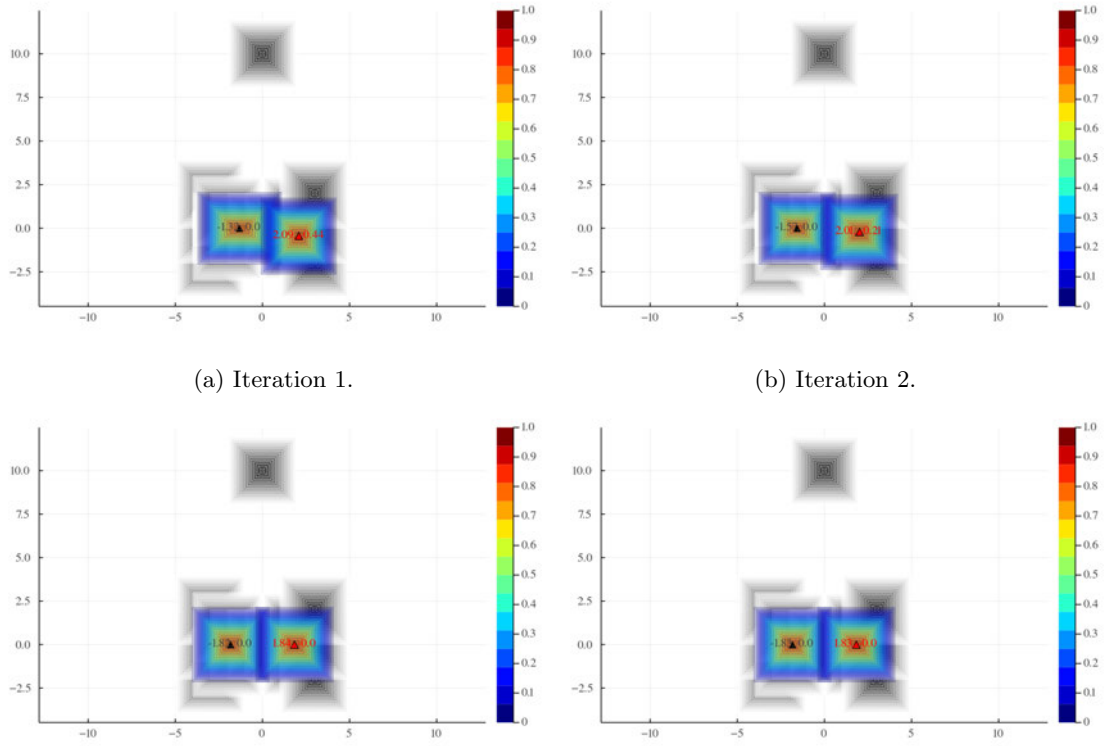
Figure 3.79: The membership functions $u_{16,1}$ and $u_{16,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{16,1}$, whereas the red functions represent $u_{16,2}$.

The LPCM with $m = 1.25$, $\eta_1 = \eta_2 = 4$ and the reflection dampening

Now let us apply the reflection dampening on the LPCM memberships with $m = 1.25$ and $\eta_1 = \eta_2 = 4$.

Figure 3.80 is the visualization of the cluster centers (computing from the reflection dampened

memberships) at different iterations. The resulting cluster centers now are much better as they are less uncertain and their lower α-cuts do not span the outlier (along the y-axis).

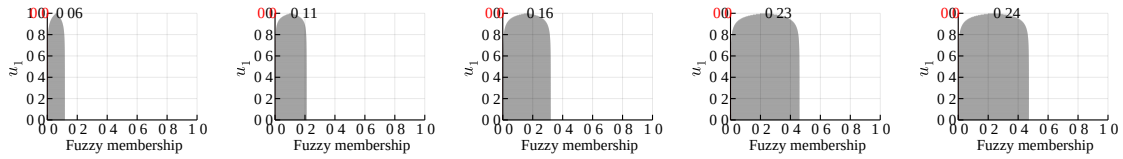


(a) Iteration 1. (b) Iteration 2.

(c) Iteration 7. (d) Iteration 13.

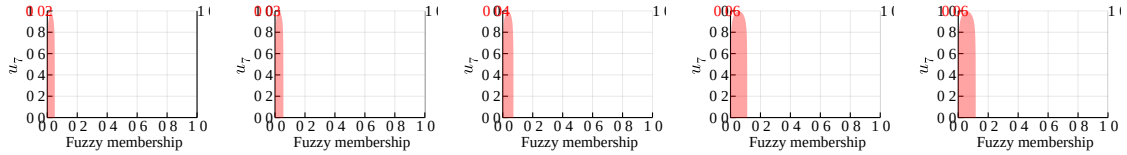
Figure 3.80: The resulting cluster centers (colored contours) from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the fuzzified butterfly+outlier2 dataset (black contours) at different iterations. The memberships used to compute these cluster centers are dampened with the reflection dampening approach.

The reflection dampened membership functions of the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ are shown in Figures 3.81–3.85. They contain much less uncertainty. Most of them are almost singleton fuzzy numbers. Note that $u_{8,1}$ has less uncertainty over time as it shifts toward "about 1" which is the effect of the reflection dampening discussed earlier.



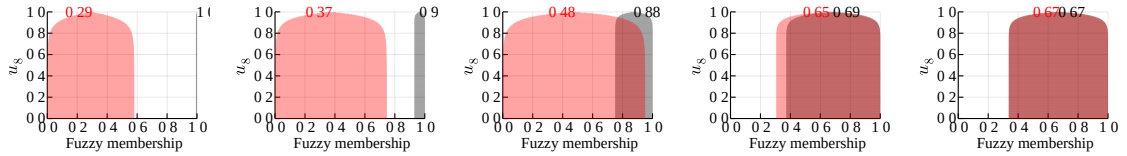
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 7. (e) Iteration 13.

Figure 3.81: The reflection dampened membership functions $u_{1,1}$ and $u_{1,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{1,1}$, whereas the red functions represent $u_{1,2}$.



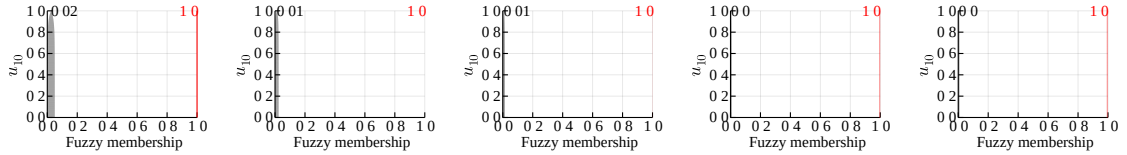
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 7. (e) Iteration 13.

Figure 3.82: The reflection dampened membership functions $u_{7,1}$ and $u_{7,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{7,1}$, whereas the red functions represent $u_{7,2}$.



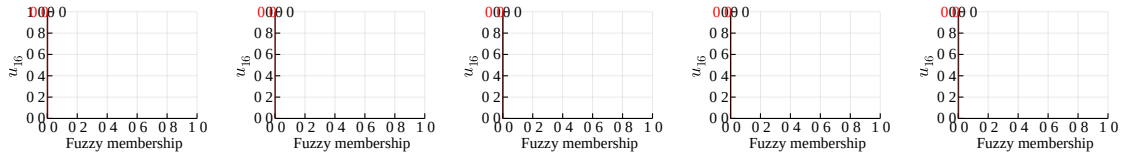
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 7. (e) Iteration 13.

Figure 3.83: The reflection dampened membership functions $u_{8,1}$ and $u_{8,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{8,1}$, whereas the red functions represent $u_{8,2}$.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 7. (e) Iteration 13.

Figure 3.84: The reflection dampened membership functions $u_{10,1}$ and $u_{10,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{10,1}$, whereas the red functions represent $u_{10,2}$.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 7. (e) Iteration 13.

Figure 3.85: The reflection dampened membership functions $u_{16,1}$ and $u_{16,2}$ from running the LPCM with $m = 1.25$ and $\eta_1 = \eta_2 = 4$ on the butterfly+outlier2 dataset at different iterations. The black functions represent $u_{16,1}$, whereas the red functions represent $u_{16,2}$.

To conclude, if the input fuzzy vectors contain high uncertainty from the beginning, it is difficult to expect the usable amount of uncertainty from LFCM and LPCM. Their memberships will be uncertain and uninformative quickly, especially for the lower level cuts. The fuzzifier m, that governs

the fuzziness on the memberships and the cluster prototypes, is also important here. As m becomes larger, the memberships are fuzzier. According to (2.42), as $m \to \infty$, the LFCM memberships approach "about $1/C$". This is also the case for LPCM, from (2.54), as m becomes larger, the LPCM memberships move toward "about 0.5".

Apart from choosing a good value of the fuzzifier m, we can also apply the dampening approaches on the memberships of LFCM or LPCM. They help us shrink the uncertainty associated with the memberships to some degree. Also, the updated cluster centers using the dampened memberships are indirectly influenced by a dampening method. As we can see from the experiments, the dampening approaches are able to reduce the uncertainty that the cluster centers contain toward the outlier, even the input fuzzy vectors are very fuzzy.

Chapter 4

CONCLUSIONS AND FUTURE RESEARCH

In this thesis, we study and investigate the LFCM and LPCM clustering algorithms. The LFCM—as an extension to the FCM—provides us a way to model uncertainty through the principle of fuzzy numbers and fuzzy vectors. Since finding a good dataset to employ these clustering algorithms is challenging, we decided to work on some synthetic datasets so that we have full control over the inputs, and also know what the expected results should be based on the results from the regular FCM and PCM. To convert the crisp butterfly dataset into the framework of fuzzy vectors, we chose the triangular fuzzy set to transform a real number in each dimension into a triangular fuzzy number. Therefore, the 1-cut (degenerate interval) is always a level cut that yields the same result as FCM and PCM. Hence, the good news here is that LFCM and LPCM generalize their ancestors, FCM and PCM.

However, we typically face the problem of membership spread in both LFCM and LPCM. That is, on the lower level cuts of the memberships, the uncertainty increases rapidly, and they will be no longer useful, i.e., they are uninformative and fill up the interval $[0, 1]$. In this work, we developed three dampening methods to mitigate the issue. According to the experiments on the synthetic datasets, they can effectively stifle the exploding uncertainty as the iterations go on. The vertical cut dampening approach is the simplest method to shrink the uncertainty. It is the same as we only consider some higher α-cuts because the α-cuts lower than β replicate the β-cut. The second dampening approach is linear dampening. It estimates the left and right slopes and dampen the estimated slopes by some degree. It always produces triangular membership functions which could be useful in terms of saving computational complexity as we can interpolate the level cuts. The last dampening method is reflection dampening that originates from the idea of preserving the original shape of a membership function. Membership functions tend to have one side (with respect to the peak) skinnier than the another side according to the membership spread problem. So, we can find

the less uncertain side and reflect across the peak. This does not only shrink the uncertainty in memberships, but also partly maintains the original original shapes of them.

The fuzzified butterfly dataset was modified by adding a new input fuzzy vector at ⟨"about 0", "about 10"⟩. It is considered as an outlier. Since the LFCM algorithm computes its memberships using the relative distances from the patterns to all cluster centers, the outlier has an influence of the LFCM memberships. The LPCM clustering algorithm that calculates its memberships/typicalities using the associated squared fuzzy distance, i.e., calculating u_{ji} using only $(d_{ji})^2$), works better when the dataset contains noise. We can see that LPCM performed well and produced cluster centers that are robust to the outlier. The coincident cluster phenomenon can happen from the LPCM if the fuzziness is high which is done by setting a larger fuzzifier m, e.g., $m = 6$.

We increased the uncertainty of the input fuzzy vectors through the widths of triangular fuzzy numbers. We can see that the uncertainty in the LFCM memberships quickly grows compared to that of the less uncertain dataset. Note that it is really important to be careful about how to fuzzify crisp patterns. Again, LPCM can result in a coincident cluster, but now much easier, because the input vectors are fuzzifer. So, finding a good value of m is also important as it has influence on the clustering results. It is even more challenging for the LPCM algorithm as it requires both the fuzzifier m and the parameters η_i. The estimation of parameters η_i is another big topic to explore.

Although, the LFCM and the LPCM clustering algorithms provide us a way to incorporate the concept of uncertainty for iterative clustering, the problems related to uncertainty need to be addressed, for example, the stopping criteria, other ways to prevent the uncertainty growth, and the parameters η_i. Estimating η_i is even more difficult in the framework of the type-2 PCM that needs more study as it is not obvious as it was for the type-1 PCM.

Furthermore, how do the LFCM and LPCM clustering algorithms perform on real datasets as opposed to synthetic datasets? It this not trival to find some good real-world datasets that do not require fuzzifying input patterns by blurring the crisp points like we did on the butterfly dataset. Future work will involve employing the type-2 clustering algorithms on a real dataset that has decent number of samples and the uncertainty is not too large.

BIBLIOGRAPHY

[1] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[2] J. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. Plenum Press, 1981.

[3] F. Rhee and C. Hwang, "A type-2 fuzzy c-means clustering algorithm," in *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*, vol. 4, 2001, pp. 1926–1929 vol.4.

[4] C. Hwang and F. C.-H. Rhee, "Uncertain fuzzy clustering: Interval type-2 fuzzy approach to c-means," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 1, pp. 107–120, 2007.

[5] E. Rubio and O. Castillo, "Interval type-2 fuzzy clustering algorithm using the combination of the fuzzy and possibilistic c-mean algorithms," in *2014 IEEE Conference on Norbert Wiener in the 21st Century (21CW)*, 2014, pp. 1–6.

[6] D. D. Nguyen, L. T. Ngo, and L. T. Pham, "Interval type-2 fuzzy c-means clustering using intuitionistic fuzzy sets," in *2013 Third World Congress on Information and Communication Technologies (WICT 2013)*, 2013, pp. 299–304.

[7] S. Auephanwiriyakul and J. Keller, "Analysis and efficient implementation of a linguistic fuzzy c-means," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 5, pp. 563–582, 2002.

[8] S. Auephanwiriyakul, "A study of linguistic pattern recognition and sensor fusion," Ph.D. dissertation, University of Missouri-Columbia, Bell & Howell Information and Learning Company, 2000.

[9] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[10] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, 1995.

[11] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*. Society for Industrial and Applied Mathematics, Jan. 2009. [Online]. Available: https://doi.org/10.1137/1.9780898717716

[12] J. M. Mendel, *Uncertain rule-based fuzzy logic system: introduction and new directions*. Prentice Hall, 2001.

[13] R. Krishnapuram and J. Keller, "A possibilistic approach to clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 98–110, 1993.

[14] E. H. Ruspini, "A new approach to clustering," *Information and Control*, vol. 15, no. 1, pp. 22–32, 1969. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019995869905919

VITA

Watchanan Chantapakul was born in Lampang, Thailand. He received the B.Eng. in computer engineering from Chiang Mai University, Chiang Mai, Thailand, in 2018. His research interests include pattern recognition, computational intelligence, and computer vision. He currently has three conference publications in print, and one conference publication has been accepted to the 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2022) at IEEE WCCI 2022.

During his time at the University of Missouri, Columbia, Watchanan served as graduate research assistant for Center for Eldercare and Rehabilitation Technology – University of Missouri. The work involved stroke patient identification from depth images.