Advances in Fuzzy Systems

Forefront of Fuzzy Logic in Data Mining: Theory, Algorithms, and Applications

Guest Editors: Gözde Ulutagay, Ronald Yager, Bernard De Baets, and Tofigh Allahviranloo

Forefront of Fuzzy Logic in Data Mining: Theory, Algorithms, and Applications

Advances in Fuzzy Systems

Forefront of Fuzzy Logic in Data Mining: Theory, Algorithms, and Applications

Guest Editors: Gözde Ulutagay, Ronald Yager, Bernard De Baets, and Tofigh Allahviranloo

Editorial Board

Contents

Forefront of Fuzzy Logic in Data Mining: Theory, Algorithms, and Applications
Gözde Ulutagay, Ronald Yager, Bernard De Baets, and Tofigh Allahviranloo
Volume 2016, Article ID 3931582, 2 pages

Cardinal Basis Piecewise Hermite Interpolation on Fuzzy Data
H. Vosoughi and S. Abbasbandy
Volume 2016, Article ID 8127215, 8 pages

Fuzzy Aspect Based Opinion Classification System for Mining Tourist Reviews
Muhammad Afzaal, Muhammad Usman, A. C. M. Fong, Simon Fong, and Yan Zhuang
Volume 2016, Article ID 6965725, 14 pages

Robust FCM Algorithm with Local and Gray Information for Image Segmentation
Hanane Barrah, Abdeljabbar Cherkaoui, and Driss Sarsri
Volume 2016, Article ID 6238295, 10 pages

Fuzzy Constrained Probabilistic Inventory Models Depending on Trapezoidal Fuzzy Numbers
Mona F. El-Wakeel and Kholood O. Al-yazidi
Volume 2016, Article ID 3673267, 10 pages

Understanding Open Source Software Evolution Using Fuzzy Data Mining Algorithm for Time Series Data
Munish Saini, Sandeep Mehmi, and Kuljit Kaur Chahal
Volume 2016, Article ID 1479692, 13 pages

An Improved Fuzzy Based Missing Value Estimation in DNA Microarray Validated by Gene Ranking
Sujay Saha, Anupam Ghosh, Dibyendu Bikash Seal, and Kashi Nath Dey
Volume 2016, Article ID 6134736, 19 pages

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2016, Article ID 3931582, 2 pages
http://dx.doi.org/10.1155/2016/3931582

Editorial

Forefront of Fuzzy Logic in Data Mining: Theory, Algorithms, and Applications

Gözde Ulutagay,[1] Ronald Yager,[2] Bernard De Baets,[3] and Tofigh Allahviranloo[4]

[1]*Department of Industrial Engineering, Izmir University, Izmir, Turkey*
[2]*Iona College, Machine Intelligence Institute, New Rochelle, NY, USA*
[3]*Department of Mathematical Modelling, Statistics and Bioinformatics, Ghent University, Ghent, Belgium*
[4]*Department of Mathematics, Islamic Azad University, Tehran, Iran*

Correspondence should be addressed to Gözde Ulutagay; gozde.ulutagay@izmir.edu.tr

Received 8 November 2016; Accepted 8 November 2016

Data mining uses various techniques and theories from a wide range of areas for the extraction of knowledge from large volumes of data. However, uncertainty is a widespread phenomenon in data mining problems. The ongoing challenges of uncertainty give rise to a plethora of knowledge extracting methods that use fuzzy logic. The aim of this special issue is twofold:

(i) to present recent outstanding developments and trends in the theory and algorithms of data mining using fuzzy logic,

(ii) to create a multidisciplinary forum for discussion on recent advances in data mining as well as new applications to biology, economics, ecology, engineering, finance, management, medicine, and so forth, using fuzzy logic.

In "Understanding Open Source Software Evolution Using Fuzzy Data Mining Algorithm for Time Series Data" by M. Saini et al., a fuzzy data mining algorithm for time series data is presented in order to generate association rules for evaluating the existing trends and regularity in the evolution of open source software projects.

In "Fuzzy Constrained Probabilistic Inventory Models Depending on Trapezoidal Fuzzy Numbers" by M. F. El-Wakeel and K. O. Al-yazidi, two types of the mixture shortage inventory model under varying order cost constraints, with lead time demand under exponential, Laplace, and uniform distributions, are discussed. Also some special cases are handled and comparisons are performed under crisp and fuzzy environments.

In "An Improved Fuzzy Based Missing Value Estimation in DNA Microarray Validated by Gene Ranking" by S. Saha et al., a modified version of the LRFDVImpute technique is proposed to impute multiple missing values of time series gene expression data. The results of imputation by a genetic algorithm (GA) based gene ranking methodology along with some regular statistical validation techniques are presented.

In "Robust FCM Algorithm with Local and Gray Information for Image Segmentation" by H. Barrah et al., a robust variant of the fuzzy c-means clustering algorithm is proposed to eliminate the drawback of its parameter dependency. The proposed algorithm is fully free of the empirical parameters and robust against noise. Moreover, a new factor that includes the local spatial and gray level information is proposed.

In "Fuzzy Aspect Based Opinion Classification System for Mining Tourist Reviews" by M. Afzaal et al., a fuzzy aspect based opinion classification system which efficiently extracts aspects from user opinions is proposed. In order to evaluate the effectiveness of the system, experiments on real world datasets are performed. According to the experimental results of the study, the proposed system is effective in aspect extraction with an improved classification accuracy.

In "Cardinal Basis Piecewise Hermite Interpolation on Fuzzy Data" by H. Vosoughi and S. Abbasbandy, interpolation of fuzzy data by the fuzzy-valued piecewise Hermite polynomial is presented for general case based on the cardinal

basis functions. Moreover, linear, cubic, and quintic situations are considered for computational examples.

We would like to thank the authors who shared their views and expertise with their excellent works, as well as the reviewers whose objective and critical comments contributed to the quality of our special issue. We strictly hope that the concerning researchers from various fields will find this special issue interesting, thought-provoking, and informative.

Gözde Ulutagay
Ronald Yager
Bernard De Baets
Tofigh Allahviranloo

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2016, Article ID 8127215, 8 pages
http://dx.doi.org/10.1155/2016/8127215

Research Article

Cardinal Basis Piecewise Hermite Interpolation on Fuzzy Data

H. Vosoughi and S. Abbasbandy

Department of Mathematics, Islamic Azad University, Science and Research Branch, Tehran, Iran

Correspondence should be addressed to S. Abbasbandy; abbasbandy@yahoo.com

Received 6 June 2016; Accepted 7 December 2016

Academic Editor: Katsuhiro Honda

A numerical method along with explicit construction to interpolation of fuzzy data through the extension principle results by widely used fuzzy-valued piecewise Hermite polynomial in general case based on the cardinal basis functions, which satisfy a vanishing property on the successive intervals, has been introduced here. We have provided a numerical method in full detail using the linear space notions for calculating the presented method. In order to illustrate the method in computational examples, we take recourse to three prime cases: linear, cubic, and quintic.

1. Introduction

Fuzzy interpolation problem was posed by Zadeh [1]. Lowen presented a solution to this problem, based on the fundamental polynomial interpolation theorem of Lagrange (see, e.g., [2]). Computational and numerical methods for calculating the fuzzy Lagrange interpolate were proposed by Kaleva [3]. He introduced an interpolating fuzzy spline of order l. Important special cases were $l = 2$, the piecewise linear interpolant, and $l = 4$, a fuzzy cubic spline. Moreover, Kaleva obtained an interpolating fuzzy cubic spline with the not-a-knot condition. Interpolating of fuzzy data was developed to simple Hermite or osculatory interpolation, $E(3)$ cubic splines, fuzzy splines, complete splines, and natural splines, respectively, in [4–8] by Abbasbandy et al. Later, Lodwick and Santos presented the Lagrange fuzzy interpolating function that loses smoothness at the knots at every α-cut; also every α-cut ($\alpha \neq 1$) of fuzzy spline with the not-α-knot boundary conditions of order k has discontinuous first derivatives on the knots and based on these interpolants some fuzzy surfaces were constructed [9]. Zeinali et al. [10] presented a method of interpolation of fuzzy data by Hermite and piecewise cubic Hermite that was simpler and consistent and also inherited smoothness properties of the generator interpolation. However, probably due to the switching points difficulties, the method was expressed in a very special case and none of three remaining important cases was not investigated and this is a fundamental reason for the method weakness.

In total, low order versions of piecewise Hermite interpolation are widely used and when we take more knots, the error breaks down uniformly to zero. Using piecewise-polynomial interpolants instead of high order polynomial interpolants on the same material and spaced knots is a useful way to diminish the wiggling and to improve the interpolation. These facts, as well as cardinal basis functions perspective, motivated us in [11] to patch cubic Hermite polynomials together to construct piecewise cubic fuzzy Hermite polynomial and provide an explicit formula in a succinct algorithm to calculate the fuzzy interpolant in cubic case as a new replacement method for [4, 10].

Now, in this paper, in light of our previous work, we want to introduce a wide general class of fuzzy-valued interpolation polynomials by extending the same approach in [11] applying a very special case of which general class of fuzzy polynomials could be an alternative to fuzzy osculatory interpolation in [4] and so its lowest order case ($m = 1$), namely, the piecewise linear polynomial, is an analogy of fuzzy linear spline in [3]. Meanwhile, when $m = 2$ with exactly the same data, we will simply produce the second lower order form of mentioned general class that was introduced in [11] and the interpolation of fuzzy data in [10].

The paper is organized in five sections. In Section 2, we have reviewed definitions and preliminary results of several

basic concepts and findings; next, we construct piecewise fuzzy Hermite polynomial in detail based on cardinal basis functions and prove some new properties of the introduced general interpolant (Section 3). In Section 4, we have produced three initial, linear, cubic [11], and quintic cases and shown the relationship between some of the mentioned cases and the newly presented interpolants in [3, 4, 10]. Furthermore, to illustrate the method, some computational examples are provided. Finally, the conclusions of this interpolation are in Section 5.

2. Preliminaries

To begin, let us introduce some brief account of notions used throughout the paper. We shall denote the set of fuzzy numbers by $\mathbb{R}_{\mathscr{F}}$ the family of all nonempty convex, normal, upper semicontinuous, and compactly supported fuzzy subsets defined on the real axis $\mathbb{R}$. Obviously, $\mathbb{R} \subset \mathbb{R}_{\mathscr{F}}$. If $u \in \mathbb{R}_{\mathscr{F}}$ is a fuzzy number, then $u^{\alpha} = \{x \in \mathbb{R} \mid u(x) \geq \alpha\}$, $0 < \alpha \leq 1$, shows the α-cut of u. For $\alpha = 0$ by the closure of the support, $u^{0} = \mathrm{cl}\{x \mid x \in \mathbb{R},\ u(x) > 0\}$. It is well known the α-cuts of $u \in \mathbb{R}_{\mathscr{F}}$ are closed bounded intervals in $\mathbb{R}$ and we will denote them by $u^{\alpha} = [\underline{u}^{\alpha}, \overline{u}^{\alpha}]$; functions $\underline{u}^{(\cdot)}, \overline{u}^{(\cdot)}$ are the lower and upper branches of u. The core of u is $u^{1} = \{x \mid x \in \mathbb{R},\ u(x) = 1\}$. In terms of α-cuts, we have the addition and the scalar multiplication:

$$\begin{aligned} (u+v)^{\alpha} &= u^{\alpha} + v^{\alpha} = \{x + y \mid x \in u^{\alpha},\ y \in v^{\alpha}\} \\ (\lambda u)^{\alpha} &= \lambda u^{\alpha} = \{\lambda x \mid x \in u^{\alpha}\} \\ (0)^{\alpha} &= \{0\} \end{aligned} \tag{1}$$

for all $0 \leq \alpha \leq 1$, $u, v \in \mathbb{R}_{\mathscr{F}}$, and $\lambda \in \mathbb{R}$.

$u = \langle a, b, c, d\rangle$ specifies a trapezoidal fuzzy number, where $a \leq b \leq c \leq d$ and if $b = c$ we obtain a triangular fuzzy number. For $\alpha \in [0, 1]$, we have $u^{\alpha} = [a+\alpha(b-a), d-\alpha(d-c)]$. In the rest of this paper, we will assume that u is a triangular fuzzy number.

Definition 1 (see, e.g., [5]). An L-R fuzzy number $u = (m, l, r)_{LR}$ is a function from the real numbers into the interval $[0, 1]$ satisfying

$$u(x) = \begin{cases} R\left(\dfrac{x-m}{r}\right) & \text{for } m \leq x \leq m + r, \\ L\left(\dfrac{m-x}{l}\right) & \text{for } m - l \leq x \leq m, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where R and L are continuous and decreasing functions from $[0, 1]$ to $[0, 1]$ fulfilling the conditions $R(0) = L(0) = 1$ and $R(1) = L(1) = 0$. When $R(x) = L(x) = 1-x$, we will have $L-L$ fuzzy numbers that involve the triangular fuzzy numbers. For an $L-L$ fuzzy number $u = (m, l, r)$, the support is the closed interval $[m - l, m + r]$ (see, e.g., [6]).

The linear space of all polynomials of degree at most N will be designated by P_N. Full Hermite interpolation problem defines a unique polynomial, called $p_N(x)$, which solves the following problem.

Theorem 2 (see [12] (existence and uniqueness)). *Let $x_0, x_1, \ldots, x_n$ be $n+1$ distinct points, $\alpha_0, \alpha_1, \ldots, \alpha_n$ be positive integers, $k = 0, 1, \ldots, \alpha_i$, and $N = \alpha_0 + \alpha_1 + \cdots + \alpha_n + n$. Set $w(x) = \prod_{i=0}^{n} (x - x_i)^{\alpha_i + 1}$ and*

$$\begin{aligned} &l_{ik}(x) \\ &= w(x) \frac{(x-x_i)^{k-\alpha_i}}{k!\,(x-x_i)^{\alpha_i+1-k}} \frac{d^{(\alpha_i-k)}}{dx^{(\alpha_i-k)}} \left[\frac{(x-x_i)^{\alpha_i+1}}{w(x)}\right]_{x=x_i} \\ &p_N(x) = \sum_{i=0}^{n} r_i l_{i0}(x) + \sum_{i=0}^{n} r_i' l_{i1}(x) + \cdots + \sum_{i=0}^{n} r_i^{(\alpha_i)} l_{i\alpha_i}(x) \end{aligned} \tag{3}$$

is a unique member of P_N for which

$$\begin{aligned} &p_N(x_0) = r_0,\ p_N'(x) = r_0', \ldots,\ p_N^{(\alpha_0-1)}(x_0) = r_0^{(\alpha_0)} \\ &\qquad\qquad \vdots \\ &p_N(x_n) = r_n,\ p_N'(x_n) = r_n', \ldots,\ p_N^{(\alpha_n-1)}(x_n) = r_n^{(\alpha_n)}. \end{aligned} \tag{4}$$

When $\alpha_0 = \alpha_1 = \cdots = \alpha_n = 1$, the full Hermite interpolation simplifies into simple Hermite or osculatory interpolation.

Definition 3. Given distinct knots $x_0, x_1, \ldots, x_n$, associated function values $f_0, f_1, \ldots, f_n$, and a linear space Φ of specific real functions generated by continuous cardinal basis functions $\phi_j : \mathbb{R} \to \mathbb{R}$, $(j = 0, 1, \ldots, n)$, $\phi_j(x_i) = \delta_{ij}$, $(i = 0, 1, \ldots, n)$, we say that the function F organized in the shape $F(x) = \sum_{j=0}^{n} f_j \phi_j(x)$ is an interpolant based on cardinal basis and such a procedure is the cardinal basis functions method.

3. Piecewise Fuzzy Hermite Interpolation Polynomial

A special case of full Hermite interpolation is piecewise Hermite interpolation (see, e.g., [13, 14]). Let us assume throughout the paper that $\Delta : a = x_0 < x_1 < \cdots < x_n = b$ is a grid of $I = [a, b]$ with knots x_i and m is a positive integer. All piecewise Hermite polynomials form a certain finite dimensional smooth linear space which we name $H_{2m-1}(\Delta; I)$.

Definition 4. $H_{2m-1}(\Delta; I)$ is a collection of all real-valued piecewise-polynomial functions $s(x)$ of degree at most $2m - 1$, defined on I, such that $s(x) \in C^{m-1}(I)$. The associated function to $s(x)$ on successive intervals $[x_{i-1}, x_i]$, $1 \leq i \leq n$, with knots from Δ, is defined by $s_i(x)$, that is, a $(m-1)$-times continuously differentiable piecewise Hermite polynomial of degree $2m - 1$, on I.

Definition 5 (see [15]). Given any real-valued function, $f(x) \in C^{m-1}(I)$. Let its unique $H_{2m-1}(\Delta; I)$-interpolate, for

m and grid Δ of I, be the element $s(x)$ of degree $2m-1$ on each interval $[x_i - 1, x_i]$, $1 \le i \le n$, such that

$$D^k s(x_i) = D^k f(x_i) \quad \forall 0 \le k \le m-1,\ 0 \le i \le n,\ D^k = \frac{d^k}{dx^k}. \tag{5}$$

Existence and uniqueness of full Hermite interpolation is provided in [12]. Because of this, presentation (5) is actually a special case of such interpolation on a gridded interval and it follows that each function belonging to $C^{m-1}(I)$ has a unique interpolate in $H_{2m-1}(\Delta; I)$.

A particular cardinal basis for linear space $H_{2m-1}(\Delta; I)$ of dimension $m(n+1)$ is $\mathscr{B} = \{\phi_{ik}(x)\}_{i=0,k=0}^{n,m-1}$, (see, e.g., [16]), where the basis function $\phi_{ik}(x)$ is defined by

$$D^l \phi_{ik}(x_j) = \delta_{kl}\delta_{ij}, \quad 0 \le k, l \le m-1,\ 0 \le i, j \le n,\ D^l = \frac{d^l}{dx^l}. \tag{6}$$

Some important results based on (6) are simple to see in the sequel, as $\phi_{i0}(x_i) = 1$ and $\phi_{i0}(x_j) = 0$ at all knots x_j and since $s(x)$ outside $[x_{i-1}, x_i]$, $1 \le i \le n$, satisfies zero data, then $\phi_{i0} \equiv 0$ for all $x_0 \le x \le x_{i-1}$ and $x_{i+1} \le x \le x_n$. $\phi_{i1}(x)$ is of degree $2m-1$, and $\phi'_{i1}(x_i) = 1$ but it is zero at all other knots. Moreover, because outside the interval $[x_{i-1}, x_{i+1}]\phi_{i1}(x)$ interpolates zero data, then $\phi_{i1}(x)$ must be vanished identically for all $x \ge x_{i+1}$ and $x \le x_{i-1}$ (see, e.g., [13, 14, 17]). Analogous reasoning applies to

$$\phi_{i1}(x) = \begin{cases} (x - x_{i-1})^m (x - x_i) \sum_{j=0}^{m-2} a_j x^j, & x_{i-1} \le x \le x_i, \\ (x - x_{i+1})^m (x - x_i) \sum_{j=0}^{m-2} b_j x^j, & x_i \le x \le x_{i+1}, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

In the following theorem, we will use the recent features.

Theorem 6. *Assume that $\phi_{ik} \in \mathscr{B}$ and satisfies the piecewise Hermite polynomial cardinal basis function constraints (6). Then,*

(i) $\phi_{i\,0}(x) + \phi_{i+1\,0}(x) \ge 1$, *for all* $x \in (x_i, x_{i+1})$, $i = 0, 1, \ldots, n-1$.

(ii) *For all* $i = 1, 2, \ldots, n-1$, ϕ_{i1} *changes the sign at* x_i. *The sign of* ϕ_{i1} *is not positive on any subinterval* $[x_{i-1}, x_i]$, *and that is not negative on* $[x_i, x_{i+1}]$.

(iii) *The sign of all other elements of* $\mathscr{B}$ *is not negative on* I.

Proof. With the assumption of (6), let $\phi_{i\,0}(x) + \phi_{i+1\,0}(x)$ be polynomial of degree $2m-1$ on the interval $[x_{i-1}, x_{i+2}]$ and interpolate the data (x_j, f_j), where $f_j = 1$ for $j = i, i+1$ and zero on the other knots of partition Δ. Suppose that $0 < i < n-1$ and $\phi_{i\,0}(x) + \phi_{i+1\,0}(x) < 1$ for some $x \in (x_i, x_{i+1})$. By the mean value theorem, its derivative has a zero on (x_i, x_{i+1}). The derivative has two $(m-2)$th order zeros at x_i and x_{i+1} and its two other zeros are x_{i-1}, x_{i+2}. Then, it has at least $2m-1$ zeros on the interval $[x_{i-1}, x_{i+2}]$, which is a contradiction. The cases $i = 0$ and $i = n-1$ are treated similarly.

In light of representation (7) and condition (6), the polynomial $\phi_{i1}(x)$ is of degree $2m-1$. It has only one minimum point on $[x_{i-1}, x_i]$ and a single maximum on the subinterval $[x_i, x_{i+1}]$. Suppose that each of the above points are one more. Then, by the mean value theorem, first derivative of $\phi_{i1}(x)$ has at least three zeros on (x_{i-1}, x_i) and three zeros on (x_i, x_{i+1}). Also, the derivative has two $(m-2)$th order zeros at x_{i-1}, x_{i+1}. Then, it has at least $2m+2$ zeros, which is a contradiction. Hence, $\phi'_{i1}(x)$ has only one zero on each of the intervals (x_{i-1}, x_i) and (x_i, x_{i+1}). Now, recall $\phi'_{i1}(x_i) = 1$; it follows that $\phi_{i1}(x) \le 0$, on $[x_{i-1}, x_i]$ and $\phi_{i1}(x) \ge 0$, on $[x_i, x_{i+1}]$. This gives (ii).

A similar proof via definition of basis functions and (6) follows the claim (iii). □

For a given $f(x) \in C^{m-1}(I)$ and its piecewise Hermite interpolate $s(x) \in H_{2m-1}(\Delta; I)$, an equivalent explicit representation of $s(x)$ in (5) can be uniquely expressed (see, e.g., [13, 17]); namely,

$$s(x) = \sum_{k=0}^{m-1} \sum_{i=0}^{n} f^{(k)}(x_i)\, \phi_{ik}(x). \tag{8}$$

Now, we want to construct a fuzzy-valued function as $s : I \to \mathbb{R}_{\mathscr{F}}$ such that $s^{(k)}(x_i) = f^{(k)}(x_i) = u_{ki} \in \mathbb{R}_{\mathscr{F}}$, $0 \le k \le m-1$, $0 \le i \le n$. Also, if for all $0 \le k \le m-1$, $0 \le i \le n$, $u_{ki} = y_i^{(k)}$ are crisp numbers in $\mathbb{R}$ and $f^{(k)}(x_i) = \chi_{y_i^{(k)}}$ (see, e.g., [2]), then there is a polynomial of degree $2m-1$ on successive intervals $[x_{i-1}, x_i]$, $0 \le i \le n$, with $s^{(k)}(x_i) = y_i^{(k)}$, $0 \le k \le m-1$, $0 \le i \le n$ such that $s(x) = \chi_{f(x)}$ for all $x \in \mathbb{R}$, where $\{(x_i, f_i, f_i', \ldots, f_i^{(m-1)}) \mid f_i^{(k)} \in \mathbb{R}_{\mathscr{F}},\ 0 \le k \le m-1,\ 0 \le i \le n\}$ is given.

We suppose that such a fuzzy function exists and we attempt to find and compute it with respect to interpolation polynomial presented by Lowen [2]. Let, for each $x \in [x_0, x_n]$, $s(x)$ be a fuzzy piecewise Hermite polynomial and $\Lambda = \{y_i^{(k)}\}_{i=0,k=0}^{n,m-1}$; then, from Kaleva [3] and Nguyen [18], we obtain the α-cuts of $s(x)$ in a succinctly algorithm as follows:

$$\begin{aligned} s^\alpha(x) &= \left\{t \in \mathbb{R} \mid \mu_{s(x)}^{(t)} \ge \alpha\right\} = \left\{t \in \mathbb{R} \mid \exists y_i^{(k)} : \mu_{u_{ki}}^{(y_i^{(k)})} \right. \\ &\ge \alpha,\ 0 \le k \le m-1,\ 0 \le i \le n,\ s_\Lambda(x) = t\Big\} = \Big\{t \\ &\in \mathbb{R} \mid t = s_\Lambda(x),\ y_i^{(k)} \in u_{ki}^\alpha,\ 0 \le k \le m-1,\ 0 \le i \\ &\le n\Big\} = \sum_{k=0}^{m-1} \sum_{i=0}^{n} u_{ki}^\alpha \phi_{ik}(x), \end{aligned} \tag{9}$$

where

$$s_\Lambda(x) = \sum_{k=0}^{m-1} \sum_{i=0}^{n} y_i^{(k)} \phi_{ik}(x) \tag{10}$$

is a piecewise Hermite polynomial in crisp case and by definition

$$s^{\alpha}(x) = \sum_{k=0}^{m-1} \sum_{i=0}^{n} u_{ki}^{\alpha} \phi_{ik}(x) \tag{11}$$

we obtain a formula that comprises a simple practical way for calculating $s(x)$:

$$s(x) = \sum_{k=0}^{m-1} \sum_{i=0}^{n} u_{ki} \phi_{ik}(x). \tag{12}$$

Since, for each $0 \le k \le m-1$, $0 \le i \le n$, $u_{ki}^{\alpha} = [\underline{u}_{ki}^{\alpha}, \overline{u}_{ki}^{\alpha}]$, then we will have $s^{\alpha}(x)$ by solving the following optimization problems:

$$\begin{aligned} &\max \ \& \ \min \quad \sum_{k=0}^{m-1} \sum_{i=0}^{n} y_i^{(k)} \phi_{ik}(x) \\ &\text{subject to} \quad \underline{u}_{ki}^{\alpha} \le y_i^{(k)} \le \overline{u}_{ki}^{\alpha}, \\ &\qquad\qquad 0 \le k \le m-1, \ 0 \le i \le n. \end{aligned} \tag{13}$$

From the ϕ_{ij}'s sign that we represented in Theorem 6, these problems have the following optimal solutions:

Maximization is as follows:

$$y_i^{(k)} = \begin{cases} \overline{u}_{ki}^{\alpha} & \text{if } \phi_{ik}(x) \ge 0 \\ \underline{u}_{ki}^{\alpha} & \text{if } \phi_{ik}(x) < 0, \end{cases} \qquad 0 \le k \le m-1, \ 0 \le i \le n. \tag{14}$$

Minimization is as follows:

$$y_i^{(k)} = \begin{cases} \underline{u}_{ki}^{\alpha} & \text{if } \phi_{ik}(x) \ge 0 \\ \overline{u}_{ki}^{\alpha} & \text{if } \phi_{ik}(x) < 0, \end{cases} \qquad 0 \le k \le m-1, \ 0 \le i \le n. \tag{15}$$

Theorem 7. *If $s(x) = \sum_{k=0}^{m-1} \sum_{i=0}^{n} u_{ki} \phi_{ik}(x)$ is an interpolating piecewise fuzzy Hermite polynomial, then for all $\alpha \in [0,1]$, $i \in \{0, 1, \ldots, n-1\}$, $x \in [x_i, x_{i+1}]$,*

$$len\, s^{\alpha}(x) \ge \min\left\{len\, s^{\alpha}(x_i), len\, s^{\alpha}(x_{i+1})\right\}, \tag{16}$$

where

$$\begin{aligned} s^{\alpha}(x) &= \left[\underline{s}^{\alpha}(x), \overline{s}^{\alpha}(x)\right], \\ len\, s^{\alpha}(x) &= \overline{s}^{\alpha}(x) - \underline{s}^{\alpha}(x). \end{aligned} \tag{17}$$

Proof. By using Theorem 6 and (11), we have $s^{\alpha}(x_i) = u_{0\,i}^{\alpha}$, $s^{\alpha}(x_{i+1}) = u_{0\,i+1}^{\alpha}$ and $\operatorname{len} s^{\alpha}(x_i) = \operatorname{len} u_{0i}^{\alpha}$, $\operatorname{len} s^{\alpha}(x_{i+1}) = \operatorname{len} u_{0\,i+1}^{\alpha}$. Since the addition does not decrease the length of an interval from (11), we can write $s^{\alpha}(x) = \sum_{k=0}^{m-1} \sum_{j=0}^{n} u_{k_j}^{\alpha} \phi_{jk}(x)$; then,

$$\begin{aligned} \operatorname{len} s^{\alpha}(x) &\ge \sum_{k=0}^{m-1} \sum_{j=0}^{n} \left|\phi_{jk}(x)\right| \operatorname{len} u_{kj}^{\alpha} \\ &\ge \sum_{j=0}^{n} \left|\phi_{j0}(x)\right| \operatorname{len} u_{0j} \\ &\ge \left|\phi_{i0}(x)\right| \operatorname{len} u_{0i} + \left|\phi_{i+1\,0}(x)\right| \operatorname{len} u_{0\,i+1} \\ &\ge \min\left\{\operatorname{len} u_{0\,i}, \operatorname{len} u_{0\,i+1}\right\} \left(\left|\phi_{i0}(x)\right| + \left|\phi_{i+1\,0}(x)\right|\right) \\ &\ge \min\left\{\operatorname{len} u_{0\,i}, \operatorname{len} u_{0\,i+1}\right\} \\ &= \min\left\{\operatorname{len} s^{\alpha}(x_i), \operatorname{len} s^{\alpha}(x_{i+1})\right\}. \end{aligned} \tag{18}$$

□

Theorem 8. *Let $u_{ki} = (m_{ki}, l_{ki}, r_{ki})$, $0 \le k \le m-1$, $0 \le i \le n$, be a triangular $L-L$ fuzzy number; then, also $s(x)$, the piecewise fuzzy Hermite polynomial interpolation, is such a fuzzy number for each x.*

Proof. The closed interval $[m-l, m+r]$ is the support of $u = (m, l, r)$, a triangular $L-L$ fuzzy number; then for each x and u_{ki}, we have

$$\begin{aligned} s(x) &= \left(\sum_{k=0}^{m-1} \sum_{i=0}^{n} u_{ki} \phi_{ik}(x)\right) \\ &= \left[\sum \sum_{\phi_{ik} \ge 0} (m_{ki} - l_{ki}) \phi_{ik}(x) \right. \\ &\quad + \sum \sum_{\phi_{ik} < 0} (m_{ki} + r_{ki}) \phi_{ik}(x), \\ &\quad \sum \sum_{\phi_{ik} \ge 0} (m_{ki} + r_{ki}) \phi_{ik}(x) \\ &\quad \left. + \sum \sum_{\phi_{ik} < 0} (m_{ki} - l_{ki}) \phi_{ik}(x)\right] = \left[\sum_{k=0}^{m-1} \sum_{i=0}^{n} m_{ki} \phi_{ik}(x)\right. \\ &\quad - \left(\sum \sum_{\phi_{ik} \ge 0} l_{ki} \phi_{ik}(x) - \sum \sum_{\phi_{ik} < 0} r_{ki} \phi_{ik}(x)\right) \\ &\quad + \sum_{k=0}^{m-1} \sum_{i=0}^{n} m_{ki} \phi_{ik}(x) \\ &\quad \left. + \left(\sum \sum_{\phi_{ik} \ge 0} r_{ki}(x) \phi_{ik}(x) - \sum \sum_{\phi_{ik} < 0} l_{ik} \phi_{ik}(x)\right)\right] \\ &= (m(x) - l(x), m(x) + r(x)). \end{aligned} \tag{19}$$

□

It follows that if $s(x) = (m(x), l(x), r(x))$, is a triangular $L - L$ fuzzy number for each x, then

$$m(x) = \sum_{k=0}^{m-1} \sum_{i=0}^{n} m_{ki}\phi_{ki},$$
$$l(x) = \sum \sum_{\phi_{ik} \geq 0} l_{ki}\phi_{ik}(x) - \sum \sum_{\phi_{ik} < 0} r_{ki}\phi_{ik}(x), \quad (20)$$
$$r(x) = \sum \sum_{\phi_{ik} \geq 0} r_{ki}\phi_{ik} - \sum \sum_{\phi_{ik} < 0} l_{ik}\phi_{ik}(x).$$

4. Piecewise-Polynomial Linear, Cubic, and Quintic Fuzzy Hermite Interpolation

We consider $m = 1$ and compute the piecewise fuzzy linear interpolant as the initial case of the presented method based on (12) and for a given set of fuzzy data $\{(x_i, f_i) \mid f_i \in \mathbb{R}_{\mathcal{F}},\ 0 \leq i \leq n\}$, as follows:

$$s(x) = \sum_{i=0}^{n} u_{0i}\phi_{i0}(x), \quad (21)$$

where $u_{0i} = f_i$, $0 \leq i \leq n$, and subject to conditions (6),

$$\phi_{00}(x) = \begin{cases} 0, & x \geq x_1 \\ \left(\dfrac{x_1 - x}{x_1 - x_0}\right), & x_0 \leq x \leq x_1 \end{cases}$$

$$\phi_{i0}(x) = \begin{cases} 0, & x \leq x_{i-1},\ x \geq x_{i+1} \\ \left(\dfrac{x - x_{i-1}}{x_i - x_{i-1}}\right), & x_{i-1} \leq x \leq x_i \\ \left(\dfrac{x_{i+1} - x}{x_{i+1} - x_i}\right), & x_i \leq x \leq x_{i+1} \end{cases} \quad (22)$$
$$1 \leq i \leq n-1$$

$$\phi_{n0}(x) = \begin{cases} 0, & x \leq x_{n-1} \\ \left(\dfrac{x - x_{n-1}}{x_n - x_{n-1}}\right), & x_{n-1} \leq x \leq x_n. \end{cases}$$

The obtained $s(x)$ is the same as fuzzy spline of order $l = 2$ that had been introduced in [3] because the basic splines and the cardinal basis functions in two interpolants are equal.

Example 9 (see [4]). Suppose the data (1, (0, 2, 1), (1, 0, 3)), (1.3, (5, 1, 2), (0, 2, 1)), (2.2, (1, 0, 3), (4, 4, 3)), (3, (4, 4, 3), (5, 1, 2)), (3.5, (0, 3, 2), (1, 1, 1)), (4, (1, 1, 1), and (0, 3, 2)). In Figure 1, the dashed line is the 0.5-cut set of piecewise cubic fuzzy interpolation $s(x)$, $x \in [1, 4]$ and the solid lines represent the support and the core of $s(x)$.

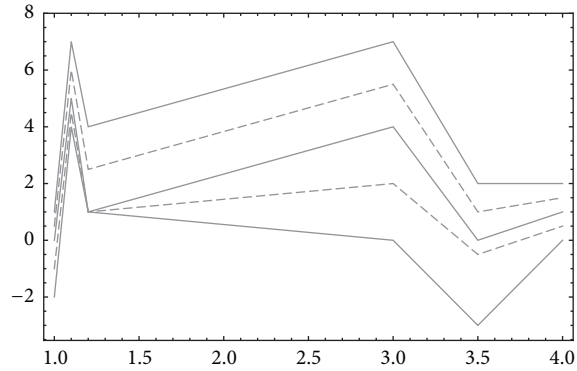

Figure 1: Graph of Example 9.

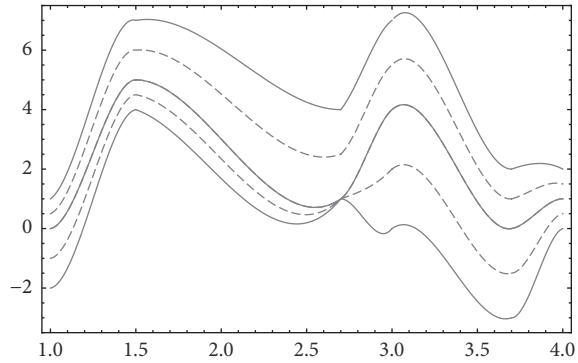

Figure 2: Graph of Example 10.

When $m = 2$, we get the piecewise cubic fuzzy Hermite polynomial interpolant in [11] for a given set of data $\{(x_i, f_i, f_i') \mid f_i, f_i' \in \mathbb{R}_{\mathcal{F}},\ 0 \leq i \leq n\}$,

$$s(x) = \sum_{i=0}^{n} u_{0i}\phi_{i0}(x) + \sum_{i=0}^{n} u_{1i}\phi_{i1}(x), \quad (23)$$

where $u_{ki} = f_i^{(k)}$, $k = 0, 1$, $0 \leq i \leq n$.

An outstanding feature of this study is that, by simply applying the second case of the presented general method and exactly the same data, we have produced an alternative to simple fuzzy Hermite polynomial interpolation in [4]. Heretofore, the mentioned cubic case (23) was independently introduced in [10] but only in very weak conditions and without using the extension principle.

The cardinal basis functions $\phi_{ik}(x)$, $k = 0, 1$, $0 \leq i \leq n$, were computed in [17].

Example 10. Suppose the data (1, (0, 2, 1), (1, 0, 3)), (1.5, (5, 1, 2), (0, 2, 1)), (2.7, (1, 0, 3), (4, 4, 3)), (3, (4, 4, 3), (5, 1, 2)), (3.7, (0, 3, 2), (1, 1, 1)), and (4, (1, 1, 1), (0, 3, 2)). In Figure 2, the dashed line is the 0.5-cut set of piecewise cubic fuzzy interpolations $s(x)$, $x \in [1, 4]$ and the solid lines represent the support and the core of $s(x)$.

Let $m = 3$; from (6), we shall construct the cardinal basis for $H_5(\Delta; I)$. The quintic Hermite polynomials

$\phi_{i0}(x)$, $\phi_{i1}(x)$, and $\phi_{i2}(x)$ are solving the interpolation problem

$$D^l\phi_{ik}\left(x_j\right) = \delta_{kl}\delta_{ij}, \quad 0 \le k, l \le 2,\ 0 \le i, j \le n. \tag{24}$$

To this end, we determine uniquely all the pervious ϕ_{ij}'s by the (24).

For $1 \le i \le n-1$, let

$$\phi_{i0}(x) = \begin{cases} \dfrac{(x_{i-1}-x)^3}{(x_{i-1}-x_i)^5}\left[(x_{i-1}+3x)(x_{i-1}-5x_i)+6x^2+10x_i^2\right], & x_{i-1} \le x \le x_i, \\ \dfrac{(x_{i+1}-x)^3}{(x_{i+1}-x_i)^5}\left[(x_{i+1}+3x)(x_{i+1}-5x_i)+6x^2+10x_i^2\right], & x_i \le x \le x_{i+1}, \\ 0, & \text{otherwise}, \end{cases}$$

$$\phi_{i1}(x) = \begin{cases} \left(\dfrac{x_{i-1}-x}{x_{i-1}-x_i}\right)^3 (x-x_i)\left[1+3\dfrac{x-x_i}{x_{i-1}-x_i}\right], & x_{i-1} \le x \le x_i, \\ \left(\dfrac{x_{i+1}-x}{x_i-x_{i+1}}\right)^3 (x_i-x)\left[1+3\dfrac{x_i-x}{x_i-x_{i+1}}\right], & x_i \le x \le x_{i+1}, \\ 0, & \text{otherwise}, \end{cases} \tag{25}$$

$$\phi_{i2}(x) = \begin{cases} \left(\dfrac{x_{i-1}-x}{x_{i-1}-x_i}\right)^3 \dfrac{(x_i-x)^2}{2}, & x_{i-1} \le x \le x_i, \\ \left(\dfrac{x-x_{i+1}}{x_i-x_{i+1}}\right)^3 \dfrac{(x_i-x)^2}{2}, & x_i \le x \le x_{i+1}, \\ 0, & \text{otherwise}. \end{cases}$$

The six next functions are similarly defined. In particular,

$$\phi_{00}(x) = \begin{cases} \dfrac{(x_1-x)^3}{(x_1-x_0)^5}\left[(x_1+3x)(x_1-5x_0)+6x^2+10x_0^3\right], & x_0 \le x \le x_1, \\ 0, & x_1 \le x \le x_n, \end{cases}$$

$$\phi_{n0}(x) = \begin{cases} \dfrac{(x_{n-1}-x)^3}{(x_{n-1}-x_n)^5}\left[(x_{n-1}+3x)(x_{n-1}-5x_n)+6x^2+10x_n^2\right], & x_{n-1} \le x \le x_n, \\ 0, & x_0 \le x \le x_{n-1}, \end{cases}$$

$$\phi_{01}(x) = \begin{cases} \left(\dfrac{x_1-x}{x_0-x_1}\right)^3 (x_0-x)\left[1+3\dfrac{x_0-x}{x_0-x_1}\right], & x_0 \le x \le x_1, \\ 0, & x_1 \le x \le x_n, \end{cases}$$

$$\phi_{n1}(x) = \begin{cases} \left(\dfrac{x_{n-1}-x}{x_{n-1}-x_n}\right)^3 (x-x_n)\left[1+3\dfrac{x-x_n}{x_{n-1}-x_n}\right], & x_{n-1} \le x \le x_n, \\ 0, & x_0 \le x \le x_{n-1}, \end{cases}$$

$$\phi_{02}(x) = \begin{cases} \left(\dfrac{x-x_1}{x_0-x_1}\right)^3 \dfrac{(x_0-x)^2}{2}, & x_0 \le x \le x_1, \\ 0, & x_1 \le x \le x_n, \end{cases}$$

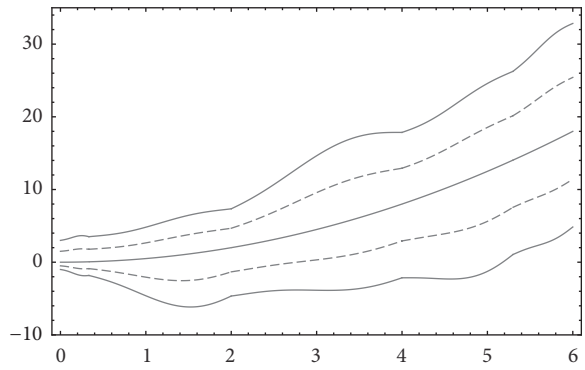

Figure 3: Graph of Example 11.

$$\phi_{n2}(x) = \begin{cases} \left(\dfrac{x_{n-1} - x}{x_{n-1} - x_n}\right)^3 \dfrac{(x_n - x)^2}{2}, & x_{n-1} \le x \le x_n, \\ 0, & x_0 \le x \le x_{n-1}. \end{cases} \tag{26}$$

Thus, we can immediately write down piecewise quintic fuzzy Hermite interpolation polynomial $s(x)$ using

$$s(x) = \sum_{i=0}^{n} u_{0i}\phi_{i0}(x) + \sum_{i=0}^{n} u_{1i}\phi_{i1}(x) + \sum_{i=0}^{n} u_{2i}\phi_{i2}(x), \tag{27}$$

where $\{(x_i, f_i, f_i', f_i'') \mid f_i^{(k)} \in \mathbb{B}_{\mathscr{F}},\ 0 \le k \le 2,\ 0 \le i \le n\}$, is given and $u_{ki} = f_i^{(k)}$.

Example 11. Suppose that (0, (0, 1, 3), (0, 2, 2), (1, 4, 4)), (1.3, (0.05, 1.9, 3.5), (0.3, 3.2, 0.8), (1, 3.1, 3)), (2, (2, 6.7, 5.3), (2, 0.5, 3.5), (1, 2.6, 2.4)), (4, (8, 10.1, 9.9), (4, 4, 0), (1, 0.6, 0.5)), (5.3, (14, 13, 12), (5.3, 0.2, 3.8), (1, 1.5, 1.7)), (6, (18, 13.2, 14.8), (6, 0.9, 3), and (1, 3.4, 3.2)) are the interpolation data. In Figure 3, the solid lines denote the support and the core of piecewise quintic fuzzy Hermite interpolation $s(x)$, $x \in [0, 6]$, and the dashed line is the 0.5-cut set of $s(x)$.

5. Conclusions and Further Work

Based on the cardinal basis functions for $m(n+1)$ dimension $H_{2m-1}(\Delta, I)$ linear space, interpolation of fuzzy data by the fuzzy-valued piecewise Hermite polynomial as the extension of same approach in [11] has been successfully introduced in general case and provided a succinct formula for calculating the new fuzzy interpolant. Moreover, two first cases of the presented method have been applied as an analogy to fuzzy spline of order two in [3] and an alternative to fuzzy osculatory interpolation in [4], respectively. In the guise of a remarkable achievement, the piecewise fuzzy cubic Hermite polynomial interpolation that was constructed with poor conditions and without using the extension principle in [10] has been produced in the role of a very special subdivision for the presented general method in this study. Finally, the third initial case, piecewise fuzzy quintic Hermite polynomial, has been described in detail. The next step to improve this method is interpolation of fuzzy data including switching points by a fuzzy differentiable piecewise interpolant.

Competing Interests

The authors declare that they have no competing interests.

References

[1] L. A. Zadeh, "Fuzzy sets," *Information and Computation*, vol. 8, pp. 338–353, 1965.

[2] R. Lowen, "A fuzzy Lagrange interpolation theorem," *Fuzzy Sets and Systems*, vol. 34, no. 1, pp. 33–38, 1990.

[3] O. Kaleva, "Interpolation of fuzzy data," *Fuzzy Sets and Systems. An International Journal in Information Science and Engineering*, vol. 61, no. 1, pp. 63–70, 1994.

[4] H. S. Goghary and S. Abbasbandy, "Interpolation of fuzzy data by Hermite polynomial," *International Journal of Computer Mathematics*, vol. 82, no. 5, pp. 595–600, 2005.

[5] H. Behforooz, R. Ezzati, and S. Abbasbandy, "Interpolation of fuzzy data by using $E(3)$ cubic splines," *International Journal of Pure and Applied Mathematics*, vol. 60, no. 4, pp. 383–392, 2010.

[6] S. Abbasbandy, R. Ezzati, and H. Behforooz, "Interpolation of fuzzy data by using fuzzy splines," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 16, no. 1, pp. 107–115, 2008.

[7] S. Abbasbandy, "Interpolation of fuzzy data by complete splines," *Korean Journal of Computational & Applied Mathematics*, vol. 8, no. 3, pp. 587–594, 2001.

[8] S. Abbasbandy and E. Babolian, "Interpolation of fuzzy data by natural splines," *The Korean Journal of Computational & Applied Mathematics*, vol. 5, no. 2, pp. 457–463, 1998.

[9] W. A. Lodwick and J. Santos, "Constructing consistent fuzzy surfaces from fuzzy data," *Fuzzy Sets and Systems. An International Journal in Information Science and Engineering*, vol. 135, no. 2, pp. 259–277, 2003.

[10] M. Zeinali, S. Shahmorad, and K. Mirnia, "Hermite and piecewise cubic Hermite interpolation of fuzzy data," *Journal of Intelligent & Fuzzy Systems*, vol. 26, no. 6, pp. 2889–2898, 2014.

[11] H. Vosoughi and S. Abbasbandy, "Interpolation of fuzzy data by cubic and piecewise-polynomial cubic hermites," *Indian Journal of Science and Technology*, vol. 9, no. 8, 2016.

[12] P. J. Davis, *Interpolation and Approximation*, Dover, New York, NY, USA, 1975.

[13] B. Wendroff, *Theoretical Numerical Analysis*, Academic Press, New York, NY, USA, 1966.

[14] P. G. Ciarlet, M. H. Schultz, and R. S. Varga, "Numerical methods of high-order accuracy for nonlinear boundary value problems," *Numerische Mathematik*, vol. 9, pp. 397–430, 1967.

[15] G. Birkhoff, M. H. Schultz, and R. S. Varga, "Piecewise Hermite interpolation in one and two variables with applications to partial differential equations," *Numerische Mathematik*, vol. 11, pp. 232–256, 1968.

[16] R. S. Varga, "Hermite interpolation-type Ritz methods for two-point boundary value problems," in *Numerical Solution of Partial Differential Equations*, J. H. Bramble, Ed., pp. 365–373, Academic Press, New York, NY, USA, 1966.

[17] P. M. Prenter, *Splines and Variational Methods*, Wiley-Interscience, 1975.

[18] H. T. Nguyen, "A note on the extension principle for fuzzy sets," *Journal of Mathematical Analysis and Applications*, vol. 64, no. 2, pp. 369–380, 1978.

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2016, Article ID 6965725, 14 pages
http://dx.doi.org/10.1155/2016/6965725

Research Article

Fuzzy Aspect Based Opinion Classification System for Mining Tourist Reviews

Muhammad Afzaal,[1] Muhammad Usman,[1] A. C. M. Fong,[2] Simon Fong,[3] and Yan Zhuang[3]

[1]*Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan*
[2]*University of Glasgow, Glasgow, UK*
[3]*University of Macau, Macau*

Correspondence should be addressed to Muhammad Usman; dr.usman@szabist-isb.edu.pk

Received 25 July 2016; Accepted 20 September 2016

Academic Editor: Gözde Ulutagay

Due to the large amount of opinions available on the websites, tourists are often overwhelmed with information and find it extremely difficult to use the available information to make a decision about the tourist places to visit. A number of opinion mining methods have been proposed in the past to identify and classify an opinion into positive or negative. Recently, aspect based opinion mining has been introduced which targets the various aspects present in the opinion text. A number of existing aspect based opinion classification methods are available in the literature but very limited research work has targeted the automatic aspect identification and extraction of implicit, infrequent, and coreferential aspects. Aspect based classification suffers from the presence of irrelevant sentences in a typical user review. Such sentences make the data noisy and degrade the classification accuracy of the machine learning algorithms. This paper presents a fuzzy aspect based opinion classification system which efficiently extracts aspects from user opinions and perform near to accurate classification. We conducted experiments on real world datasets to evaluate the effectiveness of our proposed system. Experimental results prove that the proposed system not only is effective in aspect extraction but also improves the classification accuracy.

1. Introduction

Tourism is a dynamically growing industry and important for many regions and countries as key industry [1]. Hundreds and thousands of tourists visit tourist places every year and share their opinions on various websites such as TripAdvisor and Opinion Table. These opinions give an overall view of an opinion holder regarding the tourist place [2]. However, there are a large number of opinions which are available on a particular place and it is very difficult for a normal user to review/read all these available opinions and decide on whether to visit a place or not. A number of opinion mining methods [3–13] have been proposed to deal with the large number of opinions and these methods help to classify the opinions into positive and negative. However, these previously proposed methods do not deal with various aspects present in an opinion. Instead, these methods just point out the overall expression (positive or negative) of each opinion [14]. Therefore, new aspects based opinion mining methods [15–35] were proposed. These methods allow users to extract different aspects from opinions and classify each aspect in the opinions into positive and negative. For instance in a given sentence, "Food is delicious but service is slow." The words "food" and "service" refer to aspects and "delicious" is the positive opinion of the food aspect and "slow" is the negative opinion of service aspect.

In this context, these aspects based opinion mining methods consist of two sequential tasks: (1) aspects extraction and (2) aspects based classification. Extracting aspects and classification of extracted aspects is a difficult and challenging task as reported in [8, 36, 37].

In terms of aspects extraction, firstly identifying the implicit aspects is a problem. Implicit aspects do not directly appear in any opinion but it indicates to an important aspect. For instance, in a given sentence "*Last night my wife and I visited Hasten restaurant, the taste was awesome,*" the tourist

did not mention any important aspect in this sentence. But the indication of this sentence is implying a "food" aspect.

Secondly, identifying the coreferential aspects is a difficulty. It is common that people use different words and expressions to describe the same aspect. For instance, in a restaurant opinion, atmosphere and ambience refer to the same aspect and these are coreferential to each other.

Thirdly, identifying the infrequent aspects is also very cumbersome. Due to large amount of explicit aspects available aspect extraction methods discarded the infrequent aspects. However, some infrequent aspects may be coreferential of frequent aspect or may be important for a tourist place; for instance, *Air conditioner* and *Bed* are less frequent aspects but these aspects are important for hotels.

In terms of aspects based classification, irrelevant sentences are another problem. Irrelvent sentences include self-introductory lines of opinion holders. Previous histroy of the visit adds noise and dramatically affects the accurate classification and prediction.

There is a need of an efficient aspects extraction and aspect based classification system for tourism domain to extract useful information for tourists about the different aspects of a tourist place. In this paper, we report the a new fuzzy aspect based opinion classification system. In this system, we propose a fuzzy rules based aspects extraction method that can extract explicit, implicit, and infrequent aspects and can also group coreferential aspects.

For aspect based classification we propose a three-stage fuzzy aspect based classification method using fuzzy logic algorithms. In the first stage, the opinion sentences between opinion words and aspects are filtered using Stanford Basic Dependency method proposed by [38].

In the second stage, features are built from filtered opinion sentences like *n-grams* and *Part-Of-Speech* tags. In the last stage, fuzzy logic algorithms are applied on the built features and evaluation has been performed using 10-fold cross-validation. 10-fold cross-validation is useful to limit the problems like *Overfitting* [39]. The aim of our proposed system is to extract the aspects from each opinion and to classify them into positive and negative based on the opinion words expressed in it using fuzzy logic. Fuzzy logic is handy in real life situations where decisions are taken on the basis of interlinked multiple criteria [40]. The same situation exists with aspect based opinion classification process where algorithm decides the class/label of opinions on the basis of multiple aspects and opinion words. For example, in a restaurant review, a reviewer praises the decoration of restaurant but does not approve the service provided by the staff. Hence, the decision about the opinion label, either positive or negative, is dependent on the opinion words or phrases used by the reviewer for each of the aspects.

Experiments were conducted on real world hotel and restaurant reviews taken from TripAdvisor and OpenTable websites. To evaluate the performance and effectiveness of fuzzy aspect based opinion classification system, we examined the effects of dataset size, time, feature size, feature types, and feature weighted methods on the performance of our proposed system.

In this paper, we argue that fuzzy based classification methods are very useful and effective in aspect based opinion classification. Five predominant fuzzy based algorithms, namely, Fuzzy Unordered Rule Induction Algorithm (FURIA) [41], Fuzzy Nearest Neighbors [42], Fuzzy Rough Nearest Neighbors [42], Vaguely Quantified Nearest Neighbors [42], and Fuzzy Lattice Reasoning (FLR) [43], have been compared with other similar supervised learning methods [17, 18, 20–22]. Comparison results indicate noteworthy improvement in aspect based classification. The proposed system effectively performed aspect based opinion classification, achieving an accuracy of 90.12% on restaurants dataset with FURIA and 86.02% on hotels dataset with FLR.

The rest of the paper is organized as follows. Section 2 presents an overview of previous research related to aspect based opinion mining. In Section 3, we present our proposed fuzzy based aspect extraction and classification model. It is followed by Section 4 in which experimental results on real world datasets are given. Section 5 presents the comparative evaluation of fuzzy based classification with the traditional supervised learning methods. Section 6 summarizes the contributions made in this paper.

2. Related Work

In this section, related work of aspect based opinion classification on tourism domain is presented. The purpose of this related work is to study, analyze, and identify the limitations in this area. Our overview of related topics focuses on two tasks of aspect based opinion classification: (1) aspects extraction and (2) aspect based classification.

2.1. Aspect Extraction. Aspect extraction is a major task in aspect based opinion classification. In the last few years vast majority of aspects extraction methods have been proposed for tourism domain. These methods used different ways and mechanisms to extract the important aspects from tourist reviews. We can categorize these methods into four main categories: rules based methods, seeds based methods, sequence models based methods, and topic models based methods [14].

2.1.1. Rule Based Methods. Rule based methods extracted frequent aspects from reviews using extraction rules that based on frequency, importance, appearance, and domain dependence. Extraction of frequent aspects on the basis of extraction rules is easy and effective. Pekar and Ou [15] proposed a rules based method that extracted aspects from hotels reviews using aspects appearance in each review. They applied *TermExtractor* on reviews and split them into terms and then form these terms into a lexicon. Then they manually extracted most apparent six aspects (single nouns and multiword nouns) from terms lexicon.

Similarly Muangon et al. [16] applied a *LexToPus* preprocessing on hotels reviews and split all reviews into features. These features contain both aspects and polar words. Using ranked based approach they extracted all high ranked aspects. Using the same ranked based approach de Albornoz et al. [17] also extracted high ranked nouns from hotels reviews applying a shallow preprocessing including POS tagging.

Table 1: Critical evaluation of aspects extraction methods.

Reference	Explicit aspects		Implicit aspects	Coreferential aspects	Irrelevant aspects	Method	Aspects selection	Results
	Frequent	Infrequent						
Marrese-Taylor et al., 2014 [19]	High	Null	Null	Not handled	Handled	Rules based	Frequent nouns	30%
Marrese-Taylor et al., 2013 [18]	High	Null	Null	Not handled	Handled	Rules based	Frequent nouns	Not given
de Albornoz et al., 2011 [17]	High	Null	Null	Handled	Handled	Rules based	Relative importance	66.8%
Muangon et al., 2014 [16]	High	Null	Null	Not handled	Handled	Rules based	Ranking	Not given
Pekar and Ou, 2008 [15]	High	Null	Null	Not handled	Handled	Rules based	Frequent nouns	Not given
Hai et al., 2014 [20]	High	Low	Null	Not handled	Not handled	Rules based	Domain-specific nouns	65%
Colhon et al., 2014 [21]	High	Low	Null	Not handled	Handled	Seeds based	Grammatical relationship	Not given
Mukherjee and Liu, 2012 [22]	High	Low	Null	Not handled	Handled	Seeds based	Higher-order cooccurrences	77%
Wang et al., 2010 [23]	High	Low	Null	Not handled	Handled	Seeds based	Maximum term overlapping	Not given
Zhu et al., 2011 [24]	High	Low	Null	Not handled	Handled	Seeds words	Frequency of cooccurrence	69%
Wu and Ester, 2015 [25]	High	Medium	Null	Not handled	Not handled	Topic model based	Connected topics	Not given
Xianghua et al., 2013 [26]	High	Medium	Null	Not handled	Not handled	Topic model based	Minimum distance with topics	73%
Xueke et al., 2013 [27]	High	Medium	Null	Not handled	Not handled	Topic model based	Frequent topics	Not given
Proposed method	*High*	*Medium*	*High*	*Handled*	*Handled*	*Fuzzy based*	*FURIA rules*	*81%*

We represent aspects into three types: frequent explicit aspects, infrequent explicit aspects, and implicit aspects shown in 2 to 4 columns of this table. Coreferential and irrelevant aspects handing shown in column 5 and column 6, respectively. We labeled these columns by null, low, medium, high, handled, and not handled.
"Null" = not extracting those types of aspects.
"Low" = extracting 10 to 40% of those types of aspects.
"Medium" = extracting 40 to 70% of those types of aspects.
"High" = extracting 70 to 100% of those types of aspects.
"Handled" = handling those types of aspects.
"Not handled" = not handling those types of aspects.

Marrese-Taylor et al. [18] proposed an aspect extraction algorithm to extract the aspects from restaurants reviews. They transferred all reviews sentences into POS tagged sentences using Part-Of-Speech Tagger. Later, they applied aspects extraction algorithm on POS tagged sentences. This algorithm extracted nouns that frequency is more than ten in all sentences. Similarly Marrese-Taylor et al. [19] extend [18] method. In their extended work, authors firstly used combination of POS tagging and chunking to extract aspects such as nouns and noun phrases from reviews. Secondly, they used approach frequent item set to filter out the more frequent and important aspects from all extracted aspects.

Unlike [18, 19], Hai et al. [20] proposed a different method to extract the aspects. In their method, they extracted aspects on the basis of two criteria, namely, domain-specific and not domain-independent. Firstly, they applied syntactic dependence rules to build a candidate aspects list. Secondly, calculate the score of domain-specific and domain-independent of each aspect from candidate aspects list, which they termed the intrinsic-domain relevance (IDR) score and the extrinsic domain relevance (EDR) score, respectively. Thirdly, those candidate aspects are pruned from candidate list which have low IDA and high ERD score. The limitation of the discussed rules based aspects extraction methods is that they extracted only frequent or important aspects. They pruned or discarded the aspects that have low frequent and not important for tourism domain. Table 1 categorizes these methods into four main categories: rules based methods, seeds based methods, sequence models based methods, and topic models based methods [14].

2.1.2. Seeds Based Methods. Seeds based methods extracted aspects about an tourist place from reviews using grammatical relation between seeds words with opinion words. Colhon et al. [21] selected five most debated aspects in reviews and built five seed sets of each aspect. In these seed sets each word is belonged to an important aspect. After building the seed sets they checked the grammatically relation between the

terms in reviews sentences with the each aspect seed set and then grouped these terms under this aspect.

In the same context, Mukherjee and Liu [22] grouped semantically related terms in the same aspect that are more specific and related to the seed words. Wang et al. [23] proposed an algorithm to extract the major aspects from review that is based on bootstrapping approach. In this algorithm, firstly they assigned aspect to the each sentence on the basis of maximum overlapping between sentence words and aspect. Secondly, to check the relationship between assigned aspect and sentences words they calculate the basic dependencies between them. Thirdly, which sentences words have high dependency with assigned aspect that is considered to be an aspect and added into list of aspect keywords.

Similarly Zhu et al. [24] proposed bootstrapping framework that used seed information to extract the meaningful aspects. They consider two types of terms which can be used for aspect identification: (1) POS such as nouns, adjective, adverbs, and verb and (2) n-grams. They applied C-value method on both types of terms to filter out important and meaningful terms. The meaningful terms were extracted on the basis of the frequency of occurrence of each term. As in rules based methods seed words based methods are also extracting frequent aspects but partially extracting low frequent aspects. Because list of seed words are limited that are determined by a human, with the help of these words majority of low frequent aspects could not be extracted.

2.1.3. Topic Model Based Methods. Topic model based methods are widely applied in aspects extraction and entity recognition, which is based on the assumption that each opinion is a mixture of various topics, and each topic is a probability distribution over different words. Wu and Ester [25] proposed a unified probabilistic model on users preferences about different aspects. In this model they assumed that each opinion about hotel and restaurant is connected with an aspect such as food, service, and so forth. Each of opinions described the importance of connected aspect that depends on three factors: global importance, reviewer impotence, and how much probability that aspect will be mentioned in other opinions. Based on these assumptions they used additive generative methods to extract the aspects.

Xianghua et al. [26] proposed a sliding windows based method to extract the aspects from reviews. In this method, firstly, sliding window scans the review from start to end. On each scan those words come in sliding window analyzed as aspects. Secondly, real computation process has been performed to discover the aspects accurately which were not identified in the first step.

Xu et al. [27] proposed a method JAS which adopts the classic Latent Dirichlet Allocation to make the extracted topics correspond to the reviewable aspects, rather than global properties of entities. They extracted the major aspects like food, service, and so on and fined grained aspects like staff, order, and so on of both hotels and restaurants. For the limitation of topic model based methods, they mostly governed by the phenomenon called "higher-order cooccurrence" based on how often terms cooccur in different contexts. This unfortunately results in many "nonspecific" and "irrelevant" aspects being pulled and clustered.

Moreover, there are limitations of above aspects extraction methods; they are not extracting implicit aspects and not dealing with coreferential aspects problem in reviews. Implicit aspects that do not directly appear in review but the indication of the review to a specific aspect. For example, "food" is an implicit aspect in "This restaurants taste is too good" review. In coreferential aspects problem people use different words and expressions to describe the same aspect. For example, environment and atmosphere refer to the same aspect in restaurants reviews. There should be a mechanism that categorized or grouped the similar kind of aspect.

2.2. Aspect Based Opinion Classification. Aspect based opinion classification is the determination of orientation of opinions of given text in two or more classes about aspects. Opinion classification has been performed in various classes like binary, ternary, n-ary in the form of stars, and "thumbs-up" or "thumbs-down," and so forth. We categorize the existing opinion classification of extracted aspects methods into two main categories: lexicon based and machine learning based methods.

2.2.1. Lexicon Based Methods. Lexicon based methods classified the opinions of aspects into classes using external lexicon resource. These lexicons have opinions words with the positive and negative score. Colhon et al. [21] performed binary classification of reviews using lexicon of positive and negative terms. They applied term-counting method that is based on the positive and negative terms in a review which are related to aspects of the object under discussion. In this method, a review is considered positive if it contains more positive than negative terms and negative if there are more negative than positive terms. A review is neutral if it contains (approximately) equal numbers of positive and negative terms.

Similarly Marrese-Taylor et al. [18] and Marrese-Taylor et al. [19] performed binary classification of tourist product reviews which relies on a sentiment word dictionary that contains a list of positive and negative words (called opinion words). They applied terms score method that is based on the positive and negative term scores in a review which are related to aspects. In this method, a review is considered positive if its positive terns score is greater than negative terms score, and a review is considered negative if its negative terms score is greater than positive terms score. In the same context Muangon et al. [16] performed n-ary classification of hotel review using the polar words. The term "polar words" means the lexicons which can identify the aspect such as good, bad, and expensive. This approach for extracting them from opinion text is based on syntactic pattern analysis and calculating the scores. Pekar and Ou [15] performed five-point scale classification of hotel reviews using three different lexicons. They applied opinion terms intensity method based on positive and negative opinion words score.

The limitations of lexicon based methods are domain-dependent opinions words and aspect-dependent opinion words that are used in aspect opinion classification. In domain-dependent opinion words many of the opinions

words have different opinion score in context of positive and negative in different domains. We take these two reviews "The restaurant service was very cheap" and "The price of hotel dishes was very cheap." In these reviews the "cheap" opinion word should have positive score in hotel domain and should have negative score in restaurant domain. In aspects-dependent opinion words many of the opinion words have different opinion score in different aspects. We take these two reviews "The restaurant service was very cheap" and "The price of hotel dishes was very cheap." In these reviews the "cheap" opinion word should have positive score in "price" aspect and should have negative score in "food" aspect.

2.2.2. Machine Learning Based Methods. Machine learning based methods classified the opinion of aspects into classes using different machine learning algorithms.

Wang et al. [23] proposed a novel Latent Rating Regression (LRR) method which aims to classify the opinions about aspects ratings into five-point scale. Proposed method can decompose the overall rating of a given review into ratings on different aspects and reveal the relative weights placed on those aspects by the reviewer. They implemented the proposed method on Support Vector Regression model and performed 4-fold cross-validation on hotels reviews. Results show that the proposed method acheives 78% accuracy in correctly classifying on the given dataset. In the same context Xu et al. [27] proposed aspect level opinion classification method that can predict the opinion about specific aspects ("staff," "food," and "ambience"). In order to avoid ambiguity, they only used sentences annotated with "positive" or "negative" opinion. To evaluate the model, they used the two datasets, the restaurant reviews and the hotel reviews, respectively. The restaurant reviews have been preprocessed with sentence segmentation and Part-Of-Speech tagging. For hotel reviews, they used a NLP toolkit to segment reviews into sentences and used the Stanford POS Tagger to conduct Part-Of-Speech tagging over sentences. They applied the state-of-the-art supervised learning approach, Support Vector Machine (SVM) on both datasets. They used the LibSVM to train the classifier based on the annotation information with all default options. The results show that 83.9% accuracy has been achieved using SVM classifier using 7-fold cross-validaiton.

Similarly, Pontiki et al. [28, 29] proposed system that can classify the opinions of aspect and aspect category into positive negative. They trained a SVM classifier with a linear kernel on manually labeled hotels and restaurants datasets. Then they predict the trained classifier on golden dataset that was labeled by the experts in this domain. In this system, firstly they extracted n unigram features from the respective sentences of each of the training datasets. In addition, an integer-valued feature that indicates the category of the tuple is used. The correct label for the extracted training feature vector is the corresponding polarity value (e.g., positive). Then, for each tuple {category, OTE} of a test sentence, a feature vector is built and it is classified using the trained SVM. The system scores indicate robustness across the two domains, achieving the most stable performance: 79.34% in hotels and 78.69% in restaurants.

de Albornoz et al. [17] proposed the system that aggregate the information to provide an average rating for the review. They translate the review into a Vector of Feature Intensities (VFI). A VFI is a vector of $N+1$ values, each one representing different aspects. They experience the proposed system with two strategies for assigning values to the VFI positions Binary Polarity and Probability of Polarity. In binary polarity, the aspect position is increased or decreased by 1 depending on wether the sentence was predicted as positive or negetive. In Probability of Polarity the aspect position is increased or decreased by the probability of the polarity assigned to the sentence by the polarity classifier. The VFI is used as the input to a machine learning algorithms (logistic regression, Support Vector Machine, and functional tree) that classifies the review into different rating categories. They used manually labeled hotels reviews to evaluate the method. Results indicated that 71.7% accuracy has been achieved by using logistic regression with 10-fold cross-validation.

The limitation of machine learning methods is that they need labeled data for training the classifier. Above methods use two kinds of labeled data. First is manually labeled data that are labeled by some experts who have knowledge about the domain and they assigned classes to each instance by hand. Second is automatically assigned classes data that crawled from the third party website like (TripAdvisor and Booking.com). These data have classes that have been assigned by the review owners. Manually labeled data is expensive because they need some experts to assign classes to each instance. Automatically crawled data have many useless sentences like self-introduction, previous history, and so on that diluted the opinion classification of aspects. Table 2 summarizes the limitations of lexicon based and machine learning based methods.

3. Proposed System

In this section, we describe the fuzzy aspect based opinion classification system using machine learning. The main objectives of our proposal are to extract important aspects from tourist opinions and classify each aspect opinion into positive and negative. We employ fuzzy logic based algorithms to aspects extraction and aspect based classification. Fuzzy logic algorithms are handy in aspects based opinion classification where the data is very noisy and decisions to be taken are based on multiple aspects. We utilized five predominant fuzzy logic based algorithms individual bases, FURIA, FNN, FRNN, VQNN, and FLR for determine effective of these algorithms. Figure 1 depicts the main phases of our proposed system for aspect based opinion classification. In first phase reviews are collected to build datasets about different tourist places from tourism websites. In second phase, preprocessing has been performed on collected datasets to transform reviews into sentences and eliminate the data redundancy and ambiguity in opinion words. In third phase, fuzzy rules are built using FURIA algorithm to extract and assign the aspect to each sentence of preprocessed datasets. In last phase, we performed classification on aspects assigned sentences into positive and negative using fuzzy logic algorithms.

Table 2: Critical evaluation of aspects based classification methods.

Reference	Dataset (hotels, restaurants)	Two-point scale	Five-point scale	Method	Type	Prediction	Results
Colhon et al., 2014 [21]	Review: 2521	Yes		Opinion terms counting method	Lexicon based	Compared with user reviews results	87%
Marrese-Taylor et al., 2014 [19]	Reviews: 200	Yes		Terms score method	Lexicon based	Compared with tourist experts results	90%
Pekar and Ou, 2008 [15]	Reviews: 268		Yes	Opinion terms intensity method	Lexicon based	Compared with judges results	78%
Marrese-Taylor et al., 2013 [18]	Reviews: 1435	Yes		Terms score method	Lexicon based	Compared with tourist experts results	83%
Muangon et al., 2014 [16]	Reviews: 2180	Yes		Terms score method	Lexicon based	Compared with online results	84%
Xianghua et al., 2013 [26]	Reviews: 300	Yes		Terms score method	Lexicon based	Compared with tourist experts results	75.89%
Wang et al., 2010 [23]	Reviews: 235,793		Yes	Support Vector Regression	Machine learning based	5-fold cross-validation	78%
Seki et al., 2009 [11]	Review: 1200	Yes		Naïve Bayes, SVM	Machine learning based	3-fold cross-validation	85%
Xueke et al., 2013 [27]	Reviews: 3214	Yes		Support Vector Machine	Machine learning based	7-fold cross-validation	83.9%
de Albornoz et al., 2011 [17]	Reviews: 1500	Yes		Logistic	Machine learning based	3-fold cross-validation	71.7%
Pontiki et al., 2014 [28]	Reviews: 300	Yes		Support Vector Machine	Machine learning based	3-fold cross-validation	80.15%
Pontiki et al., 2015 [29]	Reviews: 320	Yes		Maximum entropy	Machine learning based	3-fold cross-validation	78.69%
Proposed method	*Reviews: 2000 (restaurants)*	*Yes*		*FURIA*	*Machine learning based*	*10-fold cross-validation*	*90.12%*
Proposed method	*Reviews: 4000 (hotels)*	*Yes*		*FLR*	*Machine learning based*	*10-fold cross-validation*	*86.02%*

3.1. Data Collection. We collected two datasets of different sizes from restaurants and hotels domains. Restaurants dataset consists of 2000 reviews, out of which 1000 are positive and 1000 are reviews, and hostels reviews consists of 4000 reviews, out of which 2000 are positive and 2000 are reviews that we collected through a crawler from TripAdvisor website. We selected reviews of top five restaurants and top five hotels of London city from TripAdvisor website.

3.2. Data Preprocessing. The data preprocessing of collected reviews includes three steps: in the first step, we removed data redundancy because operators of the hotels and restaurants posted the review with background information about their own hotels and restaurants. If such information is included, the opinions would introduce certain bias into the dataset. So those reviews that are posted by the hotels and restaurants operators must be removed from the collected data. In the second step, we generate sentences from the collected reviews, based on sentence end characters as delimiters (i.e., period, exclamation, and question mark). In the last step, we correct ambiguous words because ambiguous words cannot be identified by classifier. Some examples of ambiguous words are "goooood, yummy, and dreamy" which are not standard English words. These words have vague meanings which may affect the aspect of opinion. So we fixed these words into standard English words like "good, yummy, and dream." After preprocessing the data, the restaurants dataset consists of 3787 sentences and hostels dataset consists of 7802 sentences.

3.3. Aspect Extraction. The aim of aspect extraction is to extract the aspects from reviews that are relevant to tourist places. We have proposed a fuzzy rules based method to extract explicit and implicit aspects from reviews. The algorithm of the proposed method is shown in Algorithm 1.

The basic workflow of the proposed aspect extraction algorithm is as follows: take all the review sentences as input then the algorithm assigns aspect to each sentence. Firstly, we extract explicit aspects from the given sentences using Stanford Part-Of-Speech Tagger [38] shown in lines (1) to (7) of Algorithm 1. In this procedure, we build Part-Of-Speech tags by applying the Tagger on each sentence shown in lines

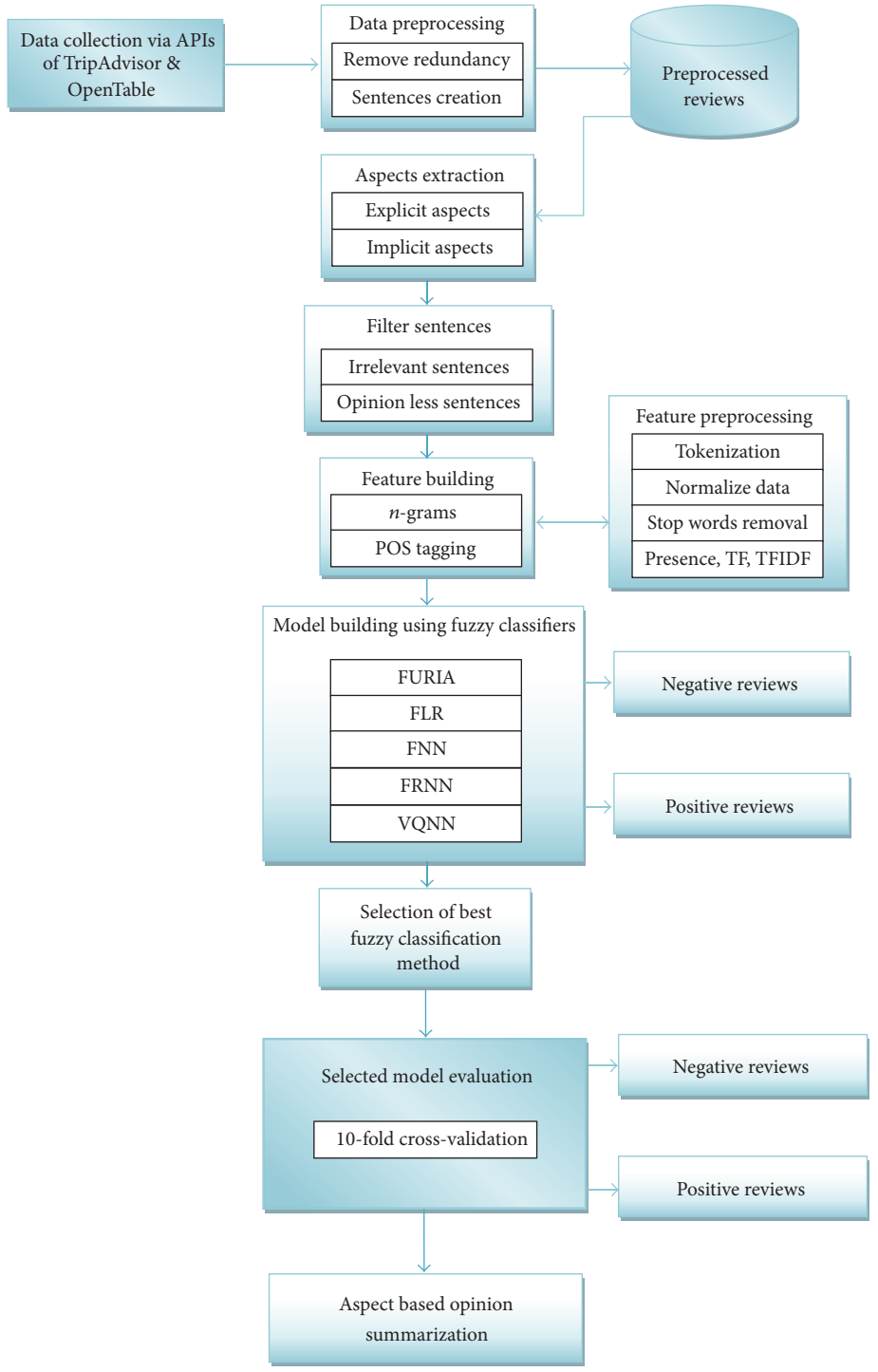

Figure 1: Proposed system for aspect based classification.

(2) and (3). Then, filter out the Noun and Noun Phrases as explicit aspects shown in lines (4) to (6). Secondly, we group all coreferential aspects that have the same meaning or indication to the same aspect by applying WordNet synonym set [44] and select high frequent one as leader of group shown in lines (8) to (14). In this procedure we match the synonyms relationship between aspects by applying WordNet on each aspect shown in lines (9) and (10). If relationship exists, then group both aspects and make the high frequent one as a leader of that group shown in lines (11) to (13). Thirdly, after extracting explicit aspects and grouping the coreferential aspects, we select the frequent aspects basis of frequency of each explicit aspect and combined frequency of each coreferential aspects group. We set ten frequencies of each frequent aspect in all sentences for selection shown in line (15). Fourthly, we build fuzzy rules using FURIA algorithm

```
Input:     Collection of sentences {S1, S2, S3, ..., Sn}
Output:    Aspects assigned to sentences
(1)   initialize aspects
(2)   for all sentences do
(3)    stanford_tagger = SPOS(sentences_i) /* Applying Stanford Part-Of-Speech Tagger on each sentence */
(4)    if NN in stanford_tagger then
(5)      aspects ← NN
(6)    end if
(7)   end for
(8)   initialize aspects_groups
(9)   for all aspects do
(10)   WordNet_sets = WNSS(aspects_i) /* Applying WordNet synonym set on each aspect */
(11)   if TRUE in WordNet_sets then
(12)     aspects_groups ← aspects_i
(13)   end if
(14)  end for
(15)  frequent_aspects = frequency_measure(aspects, group_aspets, 10) /* Filtering the frequent aspects */
(16)  fuzzy_rules = FURIA(sentences, frequent_aspects) /* Building Fuzzy rules */
(17)  initialize aspect_assigned_sentences
(18)  for all sentences do
(19)     aspect_identification = FURIA(sentences_i) /* Applying Fuzzy rules on each sentence */
(20)   if TRUE in aspect_identification then
(21)     aspect_assigned_sentences ← aspect_identification
(22)   end if
(23)  end for
(24)  return aspect_assigned_sentences
```

Algorithm 1: Aspects extraction.

on the basis of selected frequent aspects. We generate rules involving each opinion word of a sentence as the condition and a frequent aspect as the consequence, where opinion word and frequent aspect cooccur frequently in sentences shown in line (16). Fifthly, we apply the generated fuzzy rules on all the sentences to identify the aspects from all sentences and assign identified aspects to each sentence shown in lines (17) to (24). In this procedure, we match each word of sentence with built fuzzy rules shown in line (19). If match exists then assign that aspect to the sentence shown in lines (20) to (22). If match does not exist then discard that sentence from dataset. Lastly, we return all the sentences with assigned aspects shown in line (24) of Algorithm 1.

3.4. Aspect Based Opinion Classification. In aspect based classification phase, we classified the opinions of aspects into positive or negative. For this purpose we proposed fuzzy aspect based classification method that can classify the opinions of aspects into positive and negative using fuzzy logic algorithms. This method consists of three stages: filter opinion sentences, features bulling: and classifier.

3.4.1. Filter Opinion Sentences. When we crawled reviews from third part website (TripAdvisor) there were irrelevant sentences. In these irrelevant sentences, reviewer did not discuss about any aspect making it difficult to remove these sentences from reviews. There are two kinds of irrelevant sentences. One kind of irrelevant sentence exists at the start of review. Reviewers used these sentences to introduce about these trips or talked about why they visited the places, for example, "Been there a few times for lunch with friends and work but it is my first time here for dinner"; in this sentence the reviewer did not discuss about any aspect. We should remove this kind of redundant sentence that only causes the noise in the reviews. In the second kind of irrelevant sentence the reviewer just mentioned aspects but did not provide any opinion about these aspects. For example, "My father and I ordered fish, chicken and desert with the help of my uncle"; in this sentence there is no opinion in the food aspect. So they should be removed too or else they are just noise.

To remove the irreverent and aspect less sentences, we applied Stanford Basic Dependency [6] that checks the dependencies between opinion words (Adjectives) and aspects (Nouns). If an aspect does not have any dependency with opinion words then it will be removed from the reviews sentences.

3.4.2. Feature Building. The reviews sentences dataset is used to extract features that will be used to train our classifier. We built n-grams and POS tags features from the datasets. The process of obtaining n-grams and POS tags from a review is as follows: in first step of process, we tokenized the review by splitting it, on basis of spaces and punctuation marks, and form a bag of words. However, we make sure that short forms such as "don't," "I'll," and "she'd" will be considered a single word. For POS tags we extracted only verb, adverb, and adjective from dataset. In second step, we removed stop words ("a," "an," and "the") from the bag of words. In last

step, we deal with negation; a negation (such as "no" and "not") is attached to a word which precedes it or follows it. For example, a sentence "I do not like fish" will form three bigrams: "I do + not," "do + not like," "not + like fish." This last step allows improving the accuracy of the classification since the negation plays a special role in an opinion expression.

3.4.3. Classifier. Fuzzy logic algorithms are handy in real life situations where the decision to be taken is based on multiple criteria with complex interlink among them. It is very true for opinion classification process in which the algorithms must be able to understand the opinion expressed by a tourist in a review based on the opinions about various aspects of the tourist place. For example, in restaurant reviews, the some reviewers may praise the decoration of restaurant and some blame the service and staff. Deciding on the opinion as positive or negative depends on the opinion words or phrases used by the reviewers for each of the aspects. When the number of aspects is more, the complexity in the decision-making gets added and hence the decision-making becomes tough. In such situations, fuzzy logic can be effectively used. We have used five fuzzy logic based algorithms individual bases, FURIA, FNN, FRNN, VQNN, and FLR for determine effective of these algorithms in the opinion about aspects in tourist places reviews.

(1) FURIA. Fuzzy Unordered Rule Induction classifier uses greedy approach to learn rules by implementing separate and conquer strategy [41]. The classes are used to make learning rules, starting with the shortest rule. Later, all the rule instances involved in learning are removed from training data. The process is continued till all target class instances are removed [41]. The propositional version of First-Order Inductive Learner (FOIL) algorithm is used to attain the process of rule growing. An empty conjunction is assigned to the rule to initiate it and features/selectors are added till no more negative instances are covering rule. The choice of prospective feature is such that it maximizes the FOIL's information gain criterion (IG); it is a measure for rule improvement compared to default rule for the target class. This measure is given by

$$\mathrm{IG} = P_r * \left(\left(\log_2 \frac{P_r}{P_r + n_r}\right) - \left(\log_2 \frac{P}{P + N}\right)\right), \quad (1)$$

where P_r and n_r represent number of positive and negative instances participating in rule during growing phase, respectively. Similarly, P and n represent number of positive and negative instances participating in default rule, respectively. The replacement of intervals with fuzzy intervals named fuzzy set with trapezoidal membership function results in a fuzzy rule.

(2) Fuzzy Nearest Neighbors (FNN). This technique of classification is based on similarity to K nearest neighbors and these neighbors' class membership [42]. Consider a set of objects U. A test object t within U is considered to be classifying object. All remaining objects of this set U are considered to be training objects [42]. According to the algorithm, measure the fuzzy similarity of all training objects with the test object t one by one (fuzzy similarity is basically weighted distance between a training object and a classifying object) [42]. Choose K training objects having highest degrees of similarity. All of these K chosen objects have specific memberships to existing crisp classes. In simple words, each of these K chosen objects belongs to a certain class to a certain degree. Now the test object is to be classified by using the information about class membership of K nearest neighbors.

The extent $C'(t)$ to which an unclassified object t belongs to a class C is computed as follows:

$$C'(t) = \sum_{x \in N} R(x, y)\, C(x), \quad (2)$$

where $N = K$ nearest neighbors $R(x, y) = [0, 1]$-valued similarity of x and y.

(3) Fuzzy Rough Nearest Neighbors (FRNN). This algorithm combines the approach of FNN algorithm and fuzzy rough approximations [42]. From the FNN approach get nearest neighbors and from the fuzzy rough approximations get fuzzy upper and lower approximations of decision classes. For example, consider a set of objects called U [42]. One of the objects is considered test object called t and the remaining are training objects. Establish a fuzzy relation between a test object and each of the training objects. Calculate the similarity value of each couple that varies from 0 to 1. Choose the training objects with highest value of similarity as the nearest neighbors. Determine the upper and lower approximation of each class by means of the nearest neighboring objects. Predict the class membership of test object by using upper and lower approximations [42]. Output the decision class with the resulting best combined fuzzy lower and upper approximation memberships. Let D = set of decision classes, U = training data, and t = test object to be classified output class:

$N \leftarrow$ get Nearest Neighbors (t, K);
$\tau \leftarrow 0$, Class $\leftarrow \varnothing$;
$\forall C \in D$;
if $(((R \downarrow C)(t) + (R \uparrow C)(t))/2 \geq \tau)$;
Class $\leftarrow C$;
$\tau \leftarrow ((R \downarrow C)(t) + (R \uparrow C)(t))/2$.

There are two instance algorithms of FRNN named FRNN-FRS and FRNN-VQRS. Both differ in their approximations. FRNN-FRS uses traditional approximations of all and at least one; on the other hand FRNN-VQRS uses VQRS approximations of most and some. Consider a class C, so high value of upper approximation reflects that all or most of the neighboring objects belong to class C; similarly high value of lower approximation reflects that at least one or some of neighboring objects belong to class C for FRS and VQRS approximations, respectively.

(4) Vaguely Quantified Nearest Neighbors (VQNN). This algorithm is a variant of FRNN (Fuzzy Rough Nearest Neighbor) algorithm [42]. According to the algorithm a test

Table 3: Classifiers performance on restaurants and hotels dataset.

Classifier	Restaurants dataset				Hotels dataset			
	Accuracy	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure
FURIA	**90.12%**	**0.89**	**0.9**	**0.87**	79.84%	0.78	0.80	0.75
FLR	87.87%	0.87	0.88	0.87	**86.02%**	**0.85**	**0.86**	**0.85**
FNN	89.32%	0.89	0.89	0.89	75.9%	0.79	0.76	0.67
FRNN	86%	0.87	0.86	0.85	77.91%	0.82	0.77	0.78
VQNN	85.41%	0.87	0.85	0.84	75.82%	0.81	0.74	0.65

object is classified using VQRS (vaguely quantified rough set) approximations. In contrast to traditional approach, VQRS approach uses "most" and "some" quantifiers for upper and lower approximations, instead of "all" and "at least one," respectively. The VQRS approximations are favorite compared to the tradition approximations because as opposed to traditional approximations these approximations do not bring any drastic change to the upper and lower approximations by change in a single object [42]. So impact of noise will be less on VQRS approximations compared to traditional approximations. For example, consider a set of objects U containing a test object t to be classified and training objects. A subset of training objects K is considered to be the nearest neighboring objects. Classify test object to a class on basis of upper and lower approximations of a class in these nearest neighbor objects. If we consider a class C, high value of upper approximation reflects that most of the neighboring objects belong to class C; similarly high value of lower approximation reflects that some of neighboring objects belong to class C. In this algorithm, $R \uparrow C$ is replaced with $R \uparrow^{Q_l} C$ and $R \downarrow C$ is replaced with $R \downarrow^{Q_u} C$.

(5) Fuzzy Lattice Reasoning (FLR). This algorithm consists of a set of fuzzy lattice rules. These rules are induced from the training data. The classification of testing data is performed on basis of induced rules of classifier [43]. Consider that U is a set of data objects including all types of data exist in the universe but in this case focus is on lattices. A fuzzy lattice is designated as $\langle L, u\rangle$. It is couple of a lattice L and its valuation function u. It is composed of a number of its constituent elements. Each element is associated with a class [43]. Consider C as a set of classes that will be assigned to these lattice elements. Now fuzzy lattice rules are induced. Each fuzzy lattice rule is a couple of object and its corresponding class, that is, $\langle ui, ci\rangle$ that implements function $h : U \rightarrow C$. These are the couples of training set. When a new object comes in the universe then existed rules compete over it to categorize this newly coming object to a category. Consider that a new object x comes in U. Calculate its inclusion measure parameter. Now this x is presented to each rule of classifier iteratively. Rules compete to categorize x. Eventually x is categorized in one of the classes which belong to C on basis of its inclusion measure parameter.

4. System Evaluation

In this section we present the fuzzy aspect based opinion classification system evaluation experiments that determine

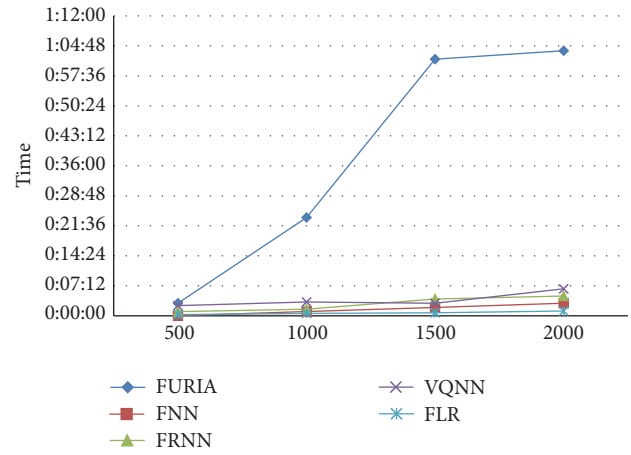

Figure 2: Classifiers prediction time with different instances sizes on restaurants dataset.

the performance of system on restaurants reviews dataset that we crawled form reviews websites such as TripAdvisor and OpenTable. We presented both tasks: (1) aspect extraction and (2) aspect based classification experiments results.

4.1. Aspects Extraction. In terms of aspects extraction, experiments are conducted to determine the percentage of correctly extracted aspects. Aspect extraction achieved a better performance 79% in restaurants dataset and 88% in hotels dataset. Results show that explicit aspects including frequent and infrequent are the most common aspects. The percentage of extracted explicit aspects for resturant and hotel datasets is 55% and 61%, respectivily. Moreover, the percentage of infrequent aspects from the explicit aspects was 21% and 23%, respectivily. Secondly, implicit aspects, the second important type, have been correctly identified from restaurant and hotel datasets with the respective percentage of 17% and 15%. Thirdly, coreferential aspects that include three or four explicit aspects, representing 7% in restaurants dataset and 10% in hotels dataset.

4.2. Aspect Based Classification. In terms of aspect based classification, experiments are conducted to examine the performance of each algorithm on different sizes of datasets, different feature weighting methods, and different feature types in aspect based classification. We also examine the time taken of each classifier on different sizes of datasets. The results obtained by aspect based classification task using restaurants dataset are presented in Table 3 and Figures 2–9. Table 3 presents the performance of each algorithm on

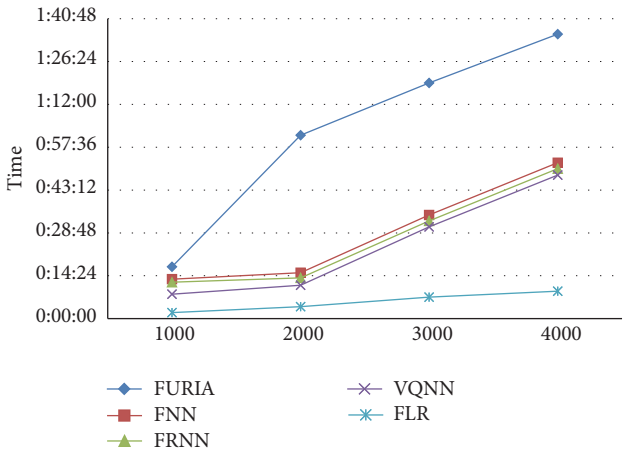

Figure 3: Classifier prediction time with different instances sizes on hotels dataset.

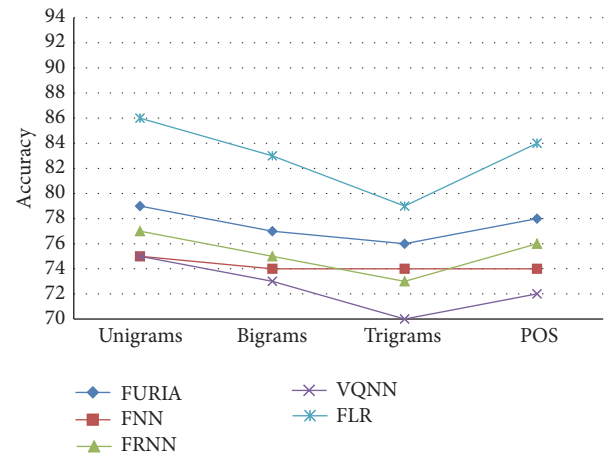

Figure 5: Classifiers performance with different feature types on hotels dataset.

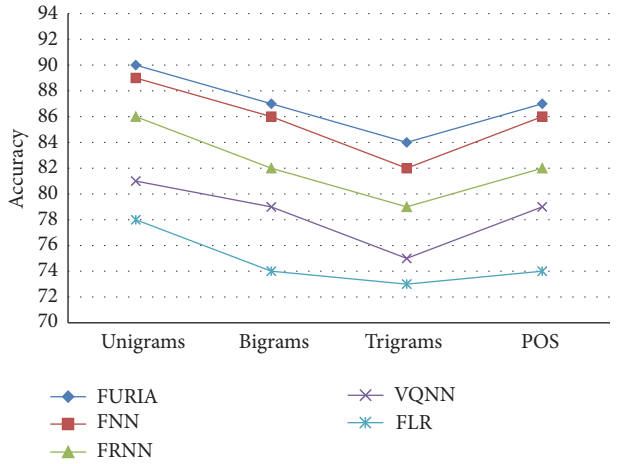

Figure 4: Classifiers performance with different feature types on restaurants dataset.

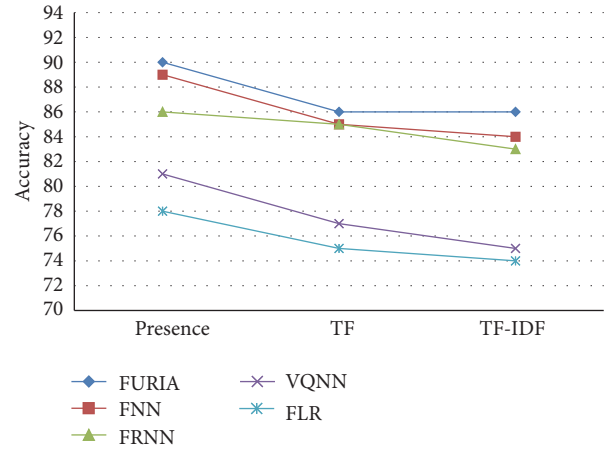

Figure 6: Classifiers performance with different feature methods on restaurants dataset.

restaurants and hotels datasets. In performance evaluation we record higher accuracy 90.12% with FURIA on restaurants dataset and 86.34% with FLR on hotels dataset.

Figures 2 and 3 present the each algorithm accuracy according to its time taken in labels prediction. In time based experiment we record that FNN takes less time in restaurants dataset and FLR takes less time in hotels dataset due to different sizes of datasets. FLR time in label prediction is quite low compared to FNN. So we can say that FLR is faster than FNN or any other fuzzy algorithms for big datasets.

Figures 4 and 5 present the effect of features types such as Unigrams, Bigrams, Trigrams, and POS on performance of aspect based classification. We run each algorithm on each feature type at the end of this experiment; we record that Unigrams and POS provide the better accuracy with FURIA on restaurants dataset and FLR on hotels dataset.

Figures 6 and 7 present the effect of feature weighting methods such as Presence, TF, and TF-IDF on performance

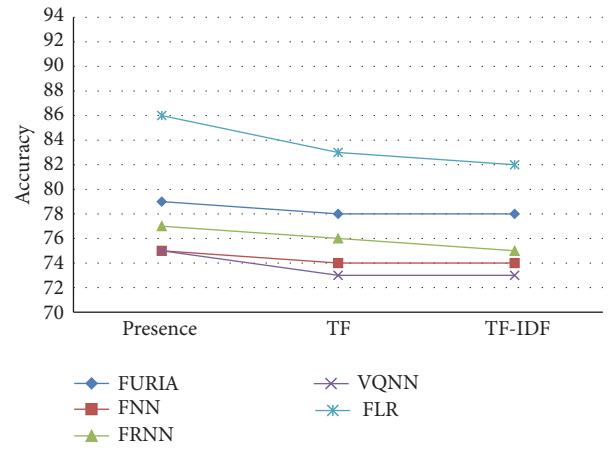

Figure 7: Classifiers performance with different feature methods on hotels dataset.

Table 4: Comparison with other aspects extraction methods.

Paper	Explicit aspects		Implicit aspects	Coreferential aspects	Irrelevant aspects	Method	Results
	Frequent	Infrequent					
de Albornoz et al., 2011 [17]	High	Null	Null	Handled	Handled	Rules based	66.8%
Mukherjee and Liu, 2012 [22]	High	Low	Null	Not handled	Handled	Seeds based	77%
Xianghua et al., 2013 [26]	High	Medium	Null	Not handled	Not handled	LDA based	73%
Proposed method (restaurants dataset)	*High*	*Medium*	*High*	*Handled*	*Handled*	*Fuzzy based*	*79%*
Proposed method (hotels dataset)	*High*	*Medium*	*High*	*Handled*	*Handled*	*Fuzzy based*	*81%*

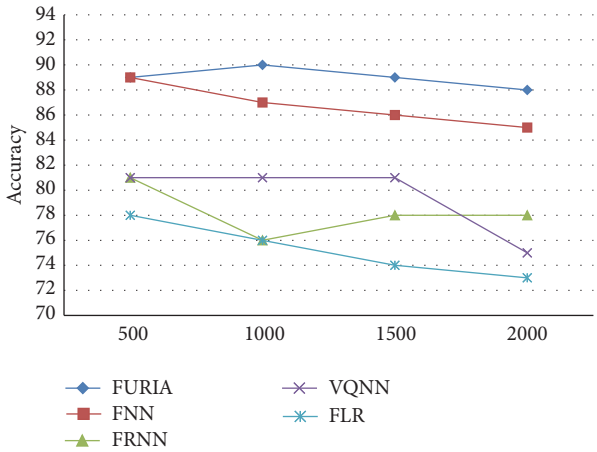

Figure 8: Classifiers performance with different instances sizes on restaurants dataset.

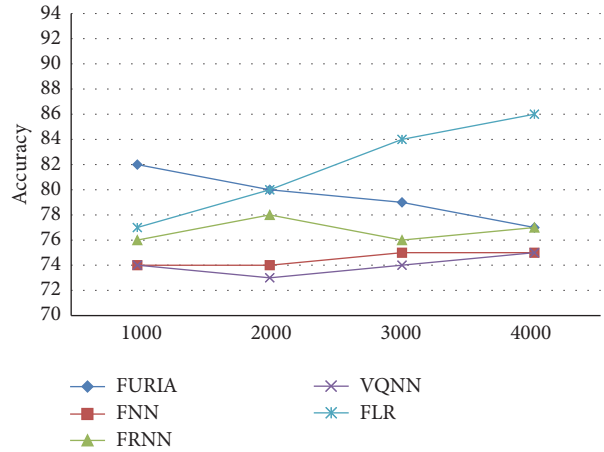

Figure 9: Classifiers performance with different instances sizes on hotels dataset.

of aspect based classification. As the previous effect of feature type experiment we use the same approach in this experiment; we run each feature weighting method on each algorithm. We record that presence weighting method provides better accuracy with FURIA on restaurants dataset and FLR on hotels dataset.

Figures 8 and 9 present that effect of size of dataset on the performance. We split both datasets into four parts for restaurants such as 500, 1000, 1500, and 2000 and for hotels such as 1000, 2000, 3000, and 4000. As mentioned above we run each algorithm on each dataset, respectively. In this experiment we record that 1000 reviews of restaurants dataset provide the better accuracy with FURIA and 4000 reviews of hotels dataset provide the better accuracy with FLR.

So the overall experiments results show that in restaurants dataset FURIA provides better accuracy and in hotels dataset FLR provides the better accuracy. On smaller dataset (restaurants reviews) where interlinking between opinion words and aspects is less FURIA build more effective rules than FLR. However, on large datasets FURIA rules are not very effective due to the complex interlinking between opinion words. FLR, on the other hand, handles such large datasets interlinkings more effectivily as compared to FURIA.

5. Comparison

In this section we compared our system with other aspects extraction and aspects based classification systems in tourism domain. Tables 4 and 5 compared the results of other systems and our proposed system. Results were computed by simply the results obtained by the best systems on tourism domain datasets. These results show an improvement, being higher than others in terms of aspects extraction and aspects based classification in the tourism domain.

6. Conclusion

In this paper, we proposed an aspect based opinion classification system that can extract aspects from reviews and classify these reviews into positive and negative about extracted aspects. Firstly in this system, we proposed fuzzy rules based method that built rules from frequent nouns and noun phrases using FURIA algorithm and used for aspects identification. Secondly, we proposed a three-stage fuzzy aspect based opinion classification method that classified the opinions of extracted aspects into positive and negative. Finally, evaluation experiments were designed to run on real world datasets taken from restaurants and hotels reviews. The

Table 5: Comparison with other aspects extraction methods.

Paper	Dataset	Method	Results
Seki et al., 2009 [11]	Review: 1200	Naïve Bayes, SVM	85%
Xueke et al., 2013 [27]	Reviews: 3214	Support Vector Machine	83.9%
de Albornoz et al., 2011 [17]	Reviews: 1500	Logistic	71.7%
Pontiki et al., 2015 [29]	Reviews: 320	Maximum entropy	78.69%
Proposed method (restaurants dataset)	*Reviews: 2000*	*FURIA*	*90.12%*
Proposed method (hotels dataset)	*Reviews: 4000*	*FLR*	*86.02%*

proposed system achieved improved results as compared to already reported results in the literature. FURIA algorithm achieved the better results as compared to other fuzzy classifiers with the 90.12% accuracy on restaurants dataset and FLR algorithm achieved the best result with the 86.02% accuracy on hotels dataset. Resultantly, tourists could easily get meaningful information about any tourist place that would be helpful to take a decision about tour to any tourist place.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

The authors of this paper are thankful to the financial supports of the grant offered with code MYRG2015-00024, called "Building Sustainable Knowledge Networks through Online Communities," by RDAO, University of Macau.

References

[1] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: tasks, approaches and applications," *Knowledge-Based Systems*, vol. 89, pp. 14–46, 2015.

[2] B. Liu, "Opinion mining and sentiment analysis," in *Web Data Mining*, pp. 459–526, Springer, New York, NY, USA, 2011.

[3] C. Bosco, V. Patti, and A. Bolioli, "Developing corpora for sentiment analysis: the case of irony and senti-TUT," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 55–63, 2013.

[4] H. Cho, S. Kim, J. Lee, and J.-S. Lee, "Data-driven integration of multiple sentiment dictionaries for lexicon-based sentiment classification of product reviews," *Knowledge-Based Systems*, vol. 71, pp. 61–71, 2014.

[5] H. Kang, S. J. Yoo, and D. Han, "Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews," *Expert Systems with Applications*, vol. 39, no. 5, pp. 6000–6010, 2012.

[6] N. Li and D. D. Wu, "Using text mining and sentiment analysis for online forums hotspot detection and forecast," *Decision Support Systems*, vol. 48, no. 2, pp. 354–368, 2010.

[7] C. Lin, Y. He, R. Everson, and S. Rüger, "Weakly supervised joint sentiment-topic detection from text," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 1134–1145, 2012.

[8] M.-T. Martín-Valdivia, E. Martínez-Cámara, J.-M. Perea-Ortega, and L. A. Ureña-López, "Sentiment polarity detection in Spanish reviews combining supervised and unsupervised approaches," *Expert Systems with Applications*, vol. 40, no. 10, pp. 3934–3942, 2013.

[9] R. Moraes, J. F. Valiati, and W. P. Gavião Neto, "Document-level sentiment classification: an empirical comparison between SVM and ANN," *Expert Systems with Applications*, vol. 40, no. 2, pp. 621–633, 2013.

[10] A. Moreo, M. Romero, J. L. Castro, and J. M. Zurita, "Lexicon-based comments-oriented news sentiment analyzer system," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9166–9180, 2012.

[11] Y. Seki, N. Kando, and M. Aono, "Multilingual opinion holder identification using author and authority viewpoints," *Information Processing & Management*, vol. 45, no. 2, pp. 189–199, 2009.

[12] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, "Sentiment classification: the contribution of ensemble learning," *Decision Support Systems*, vol. 57, no. 1, pp. 77–93, 2014.

[13] S. Tan, X. Cheng, Y. Wang, and H. Xu, "Adapting naive bayes to domain adaptation for sentiment analysis," in *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval (ECIR '09)*, pp. 337–349, Springer, Toulouse, France, 2009.

[14] L. Zhang and B. Liu, "Aspect and entity extraction for opinion mining," in *Data Mining and Knowledge Discovery for Big Data*, pp. 1–40, Springer, New York, NY, USA, 2014.

[15] V. Pekar and S. Ou, "Discovery of subjective evaluations of product features in hotel reviews," *Journal of Vacation Marketing*, vol. 14, no. 2, pp. 145–155, 2008.

[16] A. Muangon, S. Thammaboosadee, and C. Haruechaiyasak, "A lexiconizing framework of feature-based opinion mining in tourism industry," in *Proceedings of the 4th International Conference on Digital Information and Communication Technology and It's Applications (DICTAP '14)*, pp. 169–173, IEEE, Bangkok, Thailand, May 2014.

[17] J. C. de Albornoz, L. Plaza, P. Gervás, and A. Díaz, "A joint model of feature mining and sentiment analysis for product review rating," in *Advances in Information Retrieval*, pp. 55–66, Springer, Berlin, Germany, 2011.

[18] E. Marrese-Taylor, J. D. Velásquez, and F. Bravo-Marquez, "Opinion Zoom: a modular tool to explore tourism opinions on the Web," in *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT '13)*, vol. 3, IEEE Computer Society, Atlanta, Ga, USA, 2013.

[19] E. Marrese-Taylor, J. D. Velásquez, and F. Bravo-Marquez, "A novel deterministic approach for aspect-based opinion mining in tourism products reviews," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7764–7775, 2014.

[20] Z. Hai, K. Chang, J.-J. Kim, and C. C. Yang, "Identifying features in opinion mining via intrinsic and extrinsic domain relevance," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 623–634, 2014.

[21] M. Colhon, C. Bădică, and A. Şendre, "Relating the opinion holder and the review accuracy in sentiment analysis of tourist reviews," in *Knowledge Science, Engineering and Management*, pp. 246–257, Springer, Berlin, Germany, 2014.

[22] A. Mukherjee and B. Liu, "Aspect extraction through semi-supervised modeling," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1 (ACL '12)*, pp. 339–348, Association for Computational Linguistics, 2012.

[23] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis on review text data: a rating regression approach," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)*, pp. 783–792, ACM, July 2010.

[24] J. Zhu, H. Wang, M. Zhu, B. K. Tsou, and M. Ma, "Aspect-based opinion polling from customer reviews," *IEEE Transactions on Affective Computing*, vol. 2, no. 1, pp. 37–49, 2011.

[25] Y. Wu and M. Ester, "Flame: a probabilistic model combining aspect based opinion mining and collaborative filtering," in *Proceedings of the 8th ACM International Conference on Web Search and Data Mining (WSDM '15)*, ACM, Shanghai, China, 2015.

[26] F. Xianghua, L. Guo, G. Yanyan, and W. Zhiqiang, "Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon," *Knowledge-Based Systems*, vol. 37, pp. 186–195, 2013.

[27] X. Xueke, X. Cheng, S. Tan, Y. Liu, and H. Shen, "Aspect-level opinion mining of online customer reviews," *China Communications*, vol. 10, no. 3, pp. 25–41, 2013.

[28] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "Semeval-2014 task 4: aspect based sentiment analysis," in *Proceedings of the 8th international workshop on semantic evaluation (SemEval '14)*, pp. 27–35, Dublin, Ireland, 2014.

[29] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "Semeval-2015 task 12: aspect based sentiment analysis," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval '15)*, Association for Computational Linguistics, Denver, Colo, USA, 2015.

[30] Q. Su, X. Xu, H. Guo et al., "Hidden sentiment association in Chinese web opinion mining," in *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*, pp. 959–968, ACM, Beijing, China, April 2008.

[31] Z. Hai, K. Chang, and J.-J. Kim, "Implicit feature identification via co-occurrence association rule mining," in *Computational Linguistics and Intelligent Text Processing*, pp. 393–404, Springer, Berlin, Germany, 2011.

[32] G. Fei, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, "A dictionary-based approach to identifying aspects im-plied by adjectives for opinion mining," in *Proceedings of the 24th International Conference on Computational Linguistics*, p. 309, 2012.

[33] M. H. Alam, W.-J. Ryu, and S. Lee, "Joint multi-grain topic sentiment: modeling semantic aspects for online reviews," *Information Sciences*, vol. 339, pp. 206–223, 2016.

[34] X. Zheng, Z. Lin, X. Wang, K.-J. Lin, and M. Song, "Incorporating appraisal expression patterns into topic modeling for aspect and sentiment word identification," *Knowledge-Based Systems*, vol. 61, pp. 29–47, 2014.

[35] C. C. Lee and C. Hu, "Analyzing Hotel customers' E-complaints from an internet complaint forum," *Journal of Travel & Tourism Marketing*, vol. 17, no. 2-3, pp. 167–181, 2004.

[36] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 15–21, 2013.

[37] A. Mudinas, D. Zhang, and M. Levene, "Combining lexicon and learning based approaches for concept-level sentiment analysis," in *Proceedings of the 1st International Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM '12)*, ACM, Beijing, China, 2012.

[38] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1 (NAACL '03)*, pp. 173–180, Association for Computational Linguistics, Edmonton, Canada, May 2003.

[39] A. Y. Ng, "Preventing 'overfitting' of cross-validation data," in *Proceedings of the 14th International Conference on Machine Learning (ICML '97)*, pp. 245–253, 1997.

[40] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, vol. 4, Prentice Hall, Upper Saddle River, NJ, USA, 1995.

[41] J. Hühn and E. Hüllermeier, "FURIA: an algorithm for unordered fuzzy rule induction," *Data Mining and Knowledge Discovery*, vol. 19, no. 3, pp. 293–319, 2009.

[42] R. Jensen and C. Cornelis, "Fuzzy-rough nearest neighbour classification," in *Transactions on Rough Sets XIII*, pp. 56–72, Springer, Berlin, Germany, 2011.

[43] I. N. Athanasiadis, V. G. Kaburlasos, P. A. Mitkas, and V. Petridis, "Applying machine learning techniques on air quality data for real-time decision support," in *Proceedings of the 1st International NAISO Symposium on Information Technologies in Environmental Engineering (ITEE '03)*, Gdansk, Poland, 2003.

[44] G. A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2016, Article ID 6238295, 10 pages
http://dx.doi.org/10.1155/2016/6238295

Research Article

Robust FCM Algorithm with Local and Gray Information for Image Segmentation

Hanane Barrah, Abdeljabbar Cherkaoui, and Driss Sarsri

Laboratory of Innovative Technologies, National School of Applied Sciences, Tangier, Morocco

Correspondence should be addressed to Hanane Barrah; hananbarah@gmail.com

Received 25 July 2016; Revised 8 September 2016; Accepted 21 September 2016

Academic Editor: Gözde Ulutagay

The FCM (fuzzy c-mean) algorithm has been extended and modified in many ways in order to solve the image segmentation problem. However, almost all the extensions require the adjustment of at least one parameter that depends on the image itself. To overcome this problem and provide a robust fuzzy clustering algorithm that is fully free of the empirical parameters and noise type-independent, we propose a new factor that includes the local spatial and the gray level information. Actually, this work provides three extensions of the FCM algorithm that proved their efficiency on synthetic and real images.

1. Introduction

Clustering unlabeled data into the most homogeneous groups is a problem that has received extensive attention in many application domains [1–3]. Thus, several clustering methods have been developed. The hard (or crisp), probabilistic, and possibilistic c-means [4] are the well-known partitioning methods that have been extended to many different versions based on the data type and the application purpose. The probabilistic or fuzzy c-means (FCM) is always used to generate fuzzy partitions and, thus, it is widely useful to segment images [2, 5] where the fuzzy data is redundant. In fact, Abdel-Maksoud et al. [6] used the fuzzy c-means algorithm combined with its hard version k-means to extract brain tumors from MRI (Magnetic Resonance Imaging) images. In order to detect targets from radar images, Gupta [3] extended the fuzzy c-means to the fuzzy Gustafson-Kessel algorithm that uses the Mahalanobis distance instead of the Euclidean one. In addition to target detection, the proposed fuzzy Gustafson-Kessel algorithm proved its ability to clutter rejection.

Even though the standard FCM algorithm has demonstrated its accuracy in segmenting different kinds of images, it is still inefficient in the presence of noise, where its performance gradually decreases as the image noise increases. This problem is due to the lack of spatial information. To enhance the robustness and the efficiency of the standard FCM algorithm and make it strong enough in the presence of noise, lots of researchers have modified it in different ways; some have modified the objective function, while the others have used different distance metrics. In fact, Pham [7] proposed a Robust Fuzzy C-Means (RFCM) algorithm based on a generalized objective function that includes a spatial penalty on the membership function. Despite its strength in handling noisy pixels, the RFCM algorithm still suffers from many problems. First of all, the penalty term has to be computed in each iteration, which increases the computational burden. Second, the algorithm depends on a crucial parameter β that requires being selected properly in order to achieve the optimal result. Third, the spatial constraint causes a smoothing effect which can remove some fine details.

To deal with the intensity inhomogeneity in MRI images, Ahmed et al. [8] also modified the objective function of the standard FCM by including a neighborhood term that biases the labeling of a pixel by the labels of its immediate neighboring pixels. The proposed algorithm (always referred to by FCM_S) outperformed the FCM and demonstrated its usefulness in coping with "Salt and Pepper" noise. However, the FCM_S suffers from the same problems as the RFCM algorithm. In fact, the clustering accuracy depends on the selection of the parameter α that controls the tradeoff

between noise elimination and detail preservation; the spatial information causes the blurring of some fine details and computing the neighborhood term in each iteration requires the algorithm to be highly consumer in the running times point of view. To overcome this latter drawback, the FCM_S has been extended to three algorithms: The EnFCM (Enhanced FCM), FCM_S1, and FCM_S2. The first extension EnFCM was proposed by Szilágyi et al. [9] to reduce the required calculations by introducing a new factor $\gamma \in [0.5\ 1.2]$. This algorithm consists first of computing a linearly weighted sum image and then clustering it based on the gray level histogram rather than the image pixels. The segmentation quality of this algorithm is comparable to FCM_S, although the EnFCM performs quicker than its ancestors. With the same aim of making the FCM_S fast enough, Chen and Zhang [10] proposed the FCM_S1 and FCM_S2 that calculate the neighborhood term based on the mean filtered and median filtered images, respectively. As the filtered image has to be computed once and before the clustering process, the computations needed to compute the neighborhood term are drastically reduced. In fact, the authors demonstrated the effectiveness of their algorithms in artificial and real-world datasets. In [11], the authors have improved the speed of the FCM_S1 and FCM_S2 by introducing a new parameter that balances between the fastness of the hard clustering and the good quality of the fuzzy clustering. Even though the proposed algorithms have proved their fastness over the FCM_S1 and FCM_S2, they are more parameter-dependent.

By combining the main ideas of FCM_S1, FCM_S2, and EnFCM and incorporating the local spatial and the gray information together, Cai et al. [12] came up with a set of Fast Generalized Fuzzy C-Means (FGFCM) clustering algorithms. The authors proved the superiority of the FGFCM over all the aforementioned algorithms, where it overcomes the majority of their drawbacks such as controlling the tradeoff between noise-immunity and detail preserving and removing the empirically adjusted parameter α, although it requires the adjustment of a new parameter λ_g to achieve better result.

In the same context of improving the standard FCM by including the spatial information, Chuang et al. [13] proposed a fuzzy c-means algorithm that integrates the spatial information in a different way. Indeed, the authors introduced a new spatial function that is used to force the membership value of each pixel to be influenced by the membership values of its immediate neighborhood. Despite its robustness to noise and its ability to reduce the spurious blobs, this algorithm (noted by $\text{sFCM}_{p,q}$) still suffers from a major drawback where achieving the optimal segmentation requires the adjustment of two parameters p and q.

To improve the robustness to noise of the FCM_S and $\text{sFCM}_{p,q}$, Zheng et al. [14] combined their main ideas. Thus, the authors used a modified version of the spatial function proposed for $\text{sFCM}_{p,q}$ to minimize an objective function that is slightly different from the FCM_S's. The resulting algorithm surpasses all the aforementioned algorithms, but it is more parameter-dependent.

In order to deal with noise in MRI images, Ji et al. [15] proposed a Robust Spatially Constrained Fuzzy C-Means (RSCFCM) algorithm that is based on a spatial factor that works as a linear filter for smoothing and restoring noisy images. The RSCFCM algorithm minimizes a fuzzy objective function that integrates the bias field estimation, which makes it effective for intensity inhomogeneity. By testing this algorithm on synthetic and clinical images, the authors realized its better segmentation accuracy over several state-of-the-art algorithms. Nevertheless, the RSCFCM algorithm requires the adjustment of a parameter β.

So far, all the aforementioned extensions of the standard FCM have succeeded to different extents in dealing with noise. However, they all share the major drawback of adjusting empirical parameters (α, β, γ, λ_g, p, q, and h). In case of the FCM_S and its two variants FCM_S1 and FCM_S2, the parameter α controls the tradeoff between noise elimination and detail preservation. In fact, α has to be chosen large enough to remove noise and small enough to preserve fine details. Thus, the selection of this parameter is strongly dependent on the type and the level of noise. As the type and the level of noise are always a priori unknown, choosing the proper value of α remains a very difficult task, where it is always determined using trial-and-error experiments. To overcome this latter problem, this work proposes replacing the parameter α with a new factor S that includes the local spatial and the gray level information. Actually, we propose three Robust FCM algorithms: RFCMLGI (Robust FCM with Local and Gray Information), RFCMLGI_1, and RFCMLGI_2, which are direct extensions of the FCM_S, FCM_S1, and FCM_S2, respectively [10]. The proposed algorithms use the local spatial and the gray level information together to calculate the weight of the neighborhood term; the main idea here is to amplify this weight for noisy pixels and minimize it for nonnoisy ones.

In addition to the inherited advantages from FCM_S, FCM_S1, and FCM_S2, the proposed algorithms come up with valuable ones. At first, they are all fully free of the empirical parameters. Second, they control the tradeoff between noise elimination and detail preservation automatically. Third, the RFCMLGI algorithm is noise type-independent. Finally, all the algorithms are easy to be implemented, because the new factor S is proposed in a way to be easily and rapidly computed.

2. Material and Methods

The fuzzy clustering is always defined as the process of grouping, with uncertainty, unlabeled data into the most homogeneous groups or clusters as much as possible [16–18], such that the data within the same cluster are the most similar, and data from different clusters are the most dissimilar. It is an unsupervised classification, because it does not have any previous knowledge about the data structure.

2.1. Standard Fuzzy C-Means Algorithm: FCM. The fuzzy c-means or the FCM is the well-known and the best used fuzzy clustering algorithm that is based on the fuzzy sets theory [19] to create homogeneous clusters. This algorithm considers the clustering as an optimization problem where an objective function must be minimized. It receives through its input the

dataset $I = \{x_j \in \mathbb{R}^d\}_{j=1,\ldots,N}$ (part of a d-dimensional space) and the number of clusters C in order to minimize iteratively the following objective function:

$$J(I,U,V) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m} \cdot \left\|x_j - v_i\right\|^2. \quad (1)$$

$\|\cdot\|$ is the Euclidean distance, N is the number of elements in I, and m is the fuzziness exponent. $V = [v_i]$ is the set of the cluster centers. $U = [u_{ij}]$ is the fuzzy partition matrix that satisfies the following condition:

$$E = \left\{ u_{ij} \in [0,1] \mid \sum_{i=1}^{C} u_{ij} = 1,\ \forall j,\ 0 < \sum_{j=1}^{N} u_{ij} < N,\ \forall i \right\}. \quad (2)$$

The minimization of the objective function presented in (1) is carried out by updating iteratively the fuzzy partition matrix and the cluster centers as follows:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\left\|x_j - v_i\right\| / \left\|x_j - v_k\right\|\right)^{2/(m-1)}}, \quad (3)$$

$$v_i = \frac{\sum_{j=1}^{N} u_{ij}^{m} \cdot x_j}{\sum_{j=1}^{N} u_{ij}^{m}}. \quad (4)$$

Algorithm Steps

Step 0. Fix the clustering parameters (the converging error ε, the fuzziness exponent m, and the number of clusters C), input the dataset I, and initialize randomly the cluster centers.

REPEAT

Step 1. Update the partition matrix using (3).

Step 2. Update the clusters centers using (4).

UNTIL. $\|V_{\text{new}} - V_{\text{old}}\| < \varepsilon$.

V_{new} is the set of the cluster centers found in the current iteration, and V_{old} represents the previous one.

2.2. FCM with Spatial Information and Its Variants: FCM_S, FCM_S1, and FCM_S2. In order to improve the standard FCM and deal with the intensity inhomogeneities in MRI images, Ahmed et al. [8] modified the objective function (1) by introducing a neighborhood term that biases the labeling of a pixel by the labels of its immediate neighboring pixels. Thus, the authors proposed the FCM_S algorithm that minimizes the following objective function (5) using the updating functions (6) and (7) and with respect to condition E:

$$J(I,U,V) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m} \left\|x_j - v_i\right\|^2 + \frac{\alpha}{N_R}\sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m} \sum_{r\in N_j} \left\|x_r - v_i\right\|^2, \quad (5)$$

$$u_{ij} = \frac{\left(\left\|x_j - v_i\right\|^2 + (\alpha/N_R)\sum_{r\in N_j}\left\|x_r - v_i\right\|^2\right)^{-1/(m-1)}}{\sum_{k=1}^{C}\left(\left\|x_j - v_k\right\|^2 + (\alpha/N_R)\sum_{r\in N_j}\left\|x_r - v_k\right\|^2\right)^{-1/(m-1)}}, \quad (6)$$

$$v_i = \frac{\sum_{j=1}^{N} u_{ij}^{m}\left(x_j + (\alpha/N_R)\sum_{r\in N_j} x_r\right)}{(1+\alpha)\sum_{j=1}^{N} u_{ij}^{m}}. \quad (7)$$

N_j stands for the set of neighbors that exist in a window around x_j and N_R is its cardinality. The parameter α controls the effect of the neighboring term.

It is noteworthy that the neighborhood information appears in both updating functions (6) and (7), which means that the neighboring term has to be computed in each iteration; thus the FCM_S algorithm becomes very time-consuming. To get over this drawback, Chen and Zhang [10] simplified the objective function (5) to the following one:

$$J(D,U,V) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m} \cdot \left\|x_j - v_i\right\|^2 + \alpha\sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m} \cdot \left\|\overline{x}_j - v_i\right\|^2. \quad (8)$$

$\overline{x}_j$ could be the mean or the median value of the neighbors within a specified window around x_j. Actually, the authors came up with two fuzzy clustering algorithms: the FCM_S1 and FCM_S2 that use the mean filtered and median filtered images, respectively.

Like the standard FCM and FCM_S algorithms, the FCM_S1 and FCM_S2 algorithms minimize iteratively the objective function (8) by updating the fuzzy partition matrix and the cluster centers as follows:

$$u_{ij} = \frac{\left(\left\|x_j - v_i\right\|^2 + \alpha\left\|\overline{x}_j - v_i\right\|^2\right)^{-1/(m-1)}}{\sum_{k=1}^{C}\left(\left\|x_j - v_k\right\|^2 + \alpha\left\|\overline{x}_j - v_k\right\|^2\right)^{-1/(m-1)}}, \quad (9)$$

$$v_i = \frac{\sum_{j=1}^{N} u_{ij}^{m}\left(x_j + \alpha\overline{x}_j\right)}{(1+\alpha)\sum_{j=1}^{N} u_{ij}^{m}}. \quad (10)$$

Algorithm Steps

Step 0. Fix the clustering parameters (the converging error ε, the fuzziness exponent m, the number of clusters C, and the new parameter α), input the dataset I, and initialize the clusters centers randomly.

Step 1. Compute the mean (median, resp.) filtered image in case of the FCM_S1 (FCM_S2, resp.).

REPEAT

Step 2. Update the partition matrix using (9).

Step 3. Update the clusters centers using (10).

UNTIL. $\|V_{\text{new}} - V_{\text{old}}\| < \varepsilon$.

FIGURE 1: 2D square window. (a) The central pixel is noisy. (b) The central pixel is not noisy.

2.3. Robust FCM with Local and Gray Information: RFCMLGI. Even though the FCM_S, FCM_S1, and FCM_S2 have shown their strength in handling noise, adjusting the parameter α is still their major limitation. It is highly important to note that this parameter α has to be chosen large enough to eliminate noisy pixels and small enough to preserve more fine details. In other words, if the pixel under consideration is noisy, the weight of the neighboring term has to be large enough to bias the pixel's belongingness by its immediate neighborhood; if it is not, this weight has to be small enough in order not to alter significantly the pixel's belongingness and preserve it as fine detail. To respect this important note and control automatically the effect of the neighboring term, this work proposes a Robust FCM with Local and Gray Information (RFCMLGI) that is a direct extension of the FCM_S and replaces α with the new factor S defined as follows:

$$S_j = \frac{1}{N_R} \sum_{r \in N_j} \frac{\left\| x_j - x_r \right\|}{d_{rj} + 1}. \tag{11}$$

N_R and N_j are defined as in the FCM_S, and d_{rj} represents the spatial Euclidean distance between the pixels x_j and x_r.

The new factor S is calculated using the local spatial information (the spatial Euclidean distances d_{rj}) and the gray level information (the gray levels of the neighboring pixels x_r). It is defined in a way to be amplified for noisy pixels and minimized for nonnoisy ones. In fact, it is obviously deducible that S_j tends towards a maximum if x_j is noisy and its neighborhood is homogeneous, which increases the effect of the neighborhood term (see example in Figure 1(a)). Similarly, in a homogeneous window, the parameter S_j tends to a minimum, because the central pixel is not noisy; thus, the neighborhood effect decreases (see Figure 1(b)). Moreover, the contribution degree of each neighboring pixel (for calculating S_j) is inversely proportional to its spatial distance from the central pixel, which means that the nearest neighbors to the central pixel contribute more strongly than those more distant.

The RFCMLGI algorithm clusters data by minimizing iteratively the following objective function and under the previous condition E:

$$J(I,U,V) = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^{m} \cdot \left\| x_j - v_i \right\|^2 + \frac{1}{N_R} \sum_{i=1}^{C} \sum_{j=1}^{N} S_j \cdot u_{ij}^{m} \cdot \sum_{r \in N_j} \left\| x_r - v_i \right\|^2. \tag{12}$$

This optimization problem will be solved using Lagrange multiplier:

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^{m} \cdot \left\| x_j - v_i \right\|^2 + \frac{1}{N_R} \sum_{i=1}^{C} \sum_{j=1}^{N} S_j \cdot u_{ij}^{m} \cdot \sum_{r \in N_j} \left\| x_r - v_i \right\|^2 + \lambda \cdot \left(1 - \sum_{i=1}^{C} u_{ij} \right). \tag{13}$$

By taking the first derivative of J with respect to u_{ij} and setting the result to zero we find

$$\left[m \cdot u_{ij}^{m-1} \cdot \left\| x_j - v_i \right\|^2 + \frac{m \cdot S_j}{N_R} \cdot u_{ij}^{m-1} \cdot \sum_{r \in N_j} \left\| x_r - v_i \right\|^2 - \lambda \right]_{u_{ij} = u_{ij}^*} = 0. \tag{14}$$

Solving (14) for u_{ij},

$$u_{ij}^* = \left[\frac{\lambda}{m \cdot \left(\left\| x_j - v_i \right\|^2 + \left(S_j / N_R \right) \cdot \sum_{r \in N_j} \left\| x_r - v_i \right\|^2 \right)} \right]^{1/(m-1)}. \tag{15}$$

As $\sum_{i=1}^{C} u_{ij} = 1 \ \forall j \in \{1, \ldots, N\}$, then

$$\sum_{i=1}^{C} \left[\frac{\lambda}{m \cdot \left(\left\| x_j - v_i \right\|^2 + \left(S_j / N_R \right) \cdot \sum_{r \in N_j} \left\| x_r - v_i \right\|^2 \right)} \right]^{1/(m-1)} = 1. \tag{16}$$

Thus,

$$\lambda = \frac{m}{\left[\sum_{i=1}^{C} \left(1 / \left\| x_j - v_i \right\|^2 + \left(S_j / N_R \right) \cdot \sum_{r \in N_j} \left\| x_r - v_i \right\|^2 \right)^{1/(m-1)} \right]^{(m-1)}}. \tag{17}$$

Substituting λ into (15), we find

$$u_{ij}^{*} = \frac{\left(\left\|x_j - v_i\right\|^2 + \left(S_j/N_R\right)\sum_{r\in N_j}\left\|x_r - v_i\right\|^2\right)^{-1/(m-1)}}{\sum_{k=1}^{C}\left(\left\|x_j - v_k\right\|^2 + \left(S_j/N_R\right)\sum_{r\in N_j}\left\|x_r - v_k\right\|^2\right)^{-1/(m-1)}}. \quad (18)$$

This time, we take the first derivative of J with respect to v_i and setting the result to zero:

$$\left[\sum_{j=1}^{N} u_{ij}^{m}\cdot\left(x_j - v_i\right) + \frac{1}{N_R} \cdot \sum_{j=1}^{N} S_j \cdot u_{ij}^{m} \cdot \sum_{r\in N_j}\left(x_r - v_i\right)\right]_{v_i = v_i^{*}} = 0. \quad (19)$$

Solving for v_i, we find

$$v_i = \frac{\sum_{j=1}^{N} u_{ij}^{m}\cdot\left(x_j + \left(S_j/N_R\right)\sum_{r\in N_j} x_r\right)}{\sum_{j=1}^{N}\left(1 + S_j\right)\cdot u_{ij}^{m}}. \quad (20)$$

It is noticeable that the factor S is independent of the cluster centers and the membership values. Thus, its calculation takes place once at the beginning of the clustering process, which does not require much processing time. Besides, the computation of the neighborhood term is required in each iteration, which makes the algorithm much slower than the standard FCM. To overcome this last shortcoming (and by analogy to FCM_S and its two variants FCM_S1 and FCM_S2) the RFCMLGI algorithm has been extended to two simplified versions. Indeed, by simplifying the objective function presented in (12) to the following one (21) we come up with two algorithms RFCMLGI_1 and RFCMLGI_2 that update the partition matrix and the cluster centers using (22) and (23), respectively:

$$J(D,U,V) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m}\cdot\left\|x_j - v_i\right\|^2 + \sum_{i=1}^{C}\sum_{j=1}^{N} S_j\cdot u_{ij}^{m}\cdot\left\|\overline{x}_j - v_i\right\|^2, \quad (21)$$

$$u_{ij} = \frac{\left(\left\|x_j - v_i\right\|^2 + S_j\left\|\overline{x}_j - v_i\right\|^2\right)^{-1/(m-1)}}{\sum_{k=1}^{C}\left(\left\|x_j - v_k\right\|^2 + S_j\left\|\overline{x}_j - v_k\right\|^2\right)^{-1/(m-1)}}, \quad (22)$$

$$v_i = \frac{\sum_{j=1}^{N} u_{ij}^{m}\cdot\left(x_j + S_j\cdot\overline{x}_j\right)}{\sum_{j=1}^{N}\left(1 + S_j\right)\cdot u_{ij}^{m}}. \quad (23)$$

As in FCM_S1 and FCM_S2, $\overline{x}_j$ could be the mean or the median value of the neighbors within a specified window around x_j.

Algorithm Steps

Step 0. Fix the clustering parameters (the converging error ε, the fuzziness exponent m, and the number of clusters C), input the dataset I, and initialize randomly the cluster centers.

Step 1. Compute the mean (median, resp.) filtered image in case of the RFCMLGI_1 (RFCMLGI_2, resp.) algorithm.

Step 2. Compute the new factor S using (11).

REPEAT

Step 3. In case of RFCMLGI (RFCMLGI_1 or RFCMLGI_2, resp.), update the partition matrix using (18) ((22), resp.).

Step 4. In case of RFCMLGI (RFCMLGI_1 or RFCMLGI_2), update the cluster centers using (20) ((23), resp.).

UNTIL. $\|V_{\text{new}} - V_{\text{old}}\| < \varepsilon$.

The major advantages of the proposed algorithms are summarized as follows:

(i) They are fully free of the empirical parameters.

(ii) Controlling the tradeoff between noise elimination and detail preservation is automatically made.

(iii) They are easy to be implemented.

(iv) The first version of RFCMLGI is noise type-independent.

3. Results

In this section, we present some experimental results to show the efficiency of the proposed algorithms RFCMLGI, RFCMLGI_1, and RFCMLGI_2 compared to four other fuzzy clustering algorithms: FCM, FCM_S, FCM_S1, and FCM_S2. Thus, several experiments were performed on synthetic and real images and under different types and levels of noise. The clustering parameters were fixed as follows: $m = 2$, based on a study presented in [20], and $N_R = 8$ (3×3 window centered around each pixel except the central pixel).

To evaluate quantitatively the segmentation results, we use the segmentation accuracy (SA) defined as follows:

$$\text{SA} = \frac{\text{Number of correctly classified pixels}}{\text{Total number of pixels}}. \quad (24)$$

3.1. Synthetic Image. First, we apply all the algorithms to a synthetic image corrupted by different levels of Gaussian and "Salt and Pepper" noise, respectively. This image is composed of 250×250 pixels spanning into three classes with three gray level values taken as 0, 100, and 200; thus, C is fixed at 3. In these experiments, we fixed α to 4. The segmentation accuracies and results of all the algorithms are depicted in Figures 2 and 3 and Table 1, respectively.

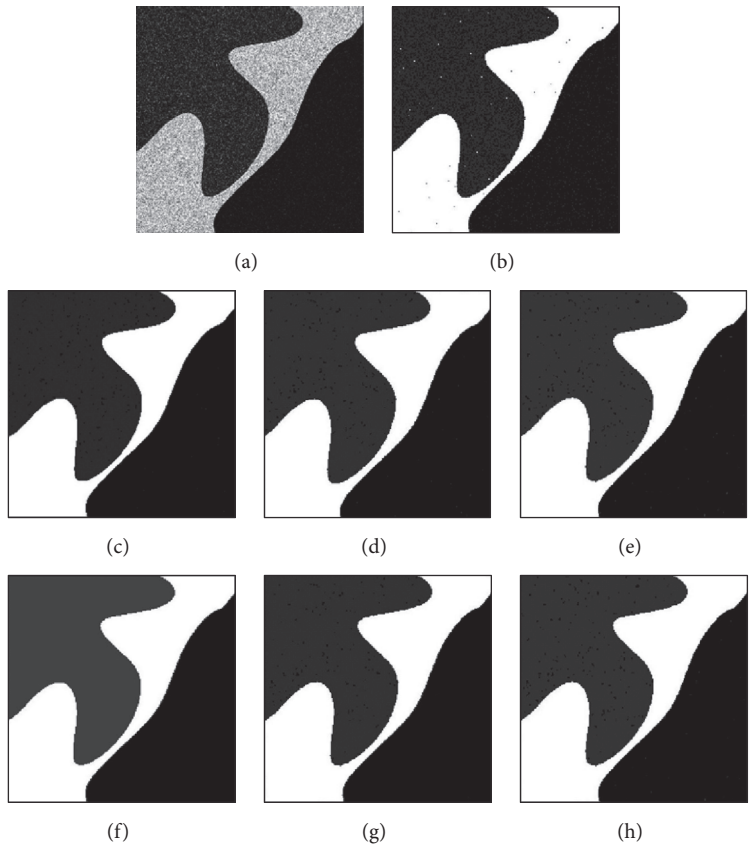

Figure 2: Segmentation results on synthetic image. (a) Original image corrupted by 15% of Gaussian noise. (b) FCM result. (c) FCM_S result. (d) FCM_S1 result. (e) FCM_S2 result. (f) RFCMLGI result. (g) RFCMLGI_1 result. (h) RFCMLGI_2 result.

From the visual results presented in Figures 2 and 3, it is clearly noticeable that all the algorithms (except the standard FCM) succeeded to different extent in handling noisy pixels. Moreover, the RFCMLGI performed better in both cases as well as the RFCMLGI_2 under Salt and Pepper noise, which is quantitatively demonstrated in Table 1.

From the numerical results depicted in Table 1, we could point out the following important notes:

(i) The segmentation accuracy decreases as the level of noise increases for all the algorithms except for the RFCMLGI under both types of noise and RFCMLGI_2 under Salt and Pepper noise.

(ii) For each type and level of noise, the proposed algorithms RFCMLGI, RFCMLGI_1, and RFCMLGI_2 outperformed the FCM_S, FCM_S1, and FCM_S2, respectively. And, more specifically, the segmentation accuracies produced by the RFCMLGI are more or less similar, which means that this algorithm is less dependent on the noise type.

(iii) Under Salt and Pepper noise, the segmentation accuracies performed by the RFCMLGI_2 are equal and tend towards the maximum, which proves the convenience of this algorithm to segment images corrupted by Salt and Pepper noise.

(iv) Under Gaussian noise, RFCMLGI has the best performance.

Based on the previous remarks, we conclude that the proposed algorithms surpassed the FCM_S and its two variants. In addition, if the type of noise is unknown the RFCMLGI is the best choice.

3.2. Real Images. To validate our methods, we test them on two real images and compare their results with the best results of the FCM_S, FCM_S1, and FCM_S2 that are obtained by seeking the value of α after which there are no apparent changes in the segmentation accuracy.

3.2.1. Selection of α. We use the trial-and-error method to select the best values of α to segment the previous synthetic image corrupted by 30% of Gaussian and Salt and Pepper noise, respectively. Actually, under Gaussian (Salt and Pepper, resp.) noise and for $\alpha \geq 10.2$ ($\alpha \geq 6.8$, resp.) there are no apparent changes in the segmentation accuracy of the FCM_S and FCM_S1 (FCM_S2, resp.).

3.2.2. Eight Image. This image was corrupted by 30% of Gaussian and Salt and Pepper noise, respectively. Even though this image contains two objects, we fixed C to 3; the third cluster

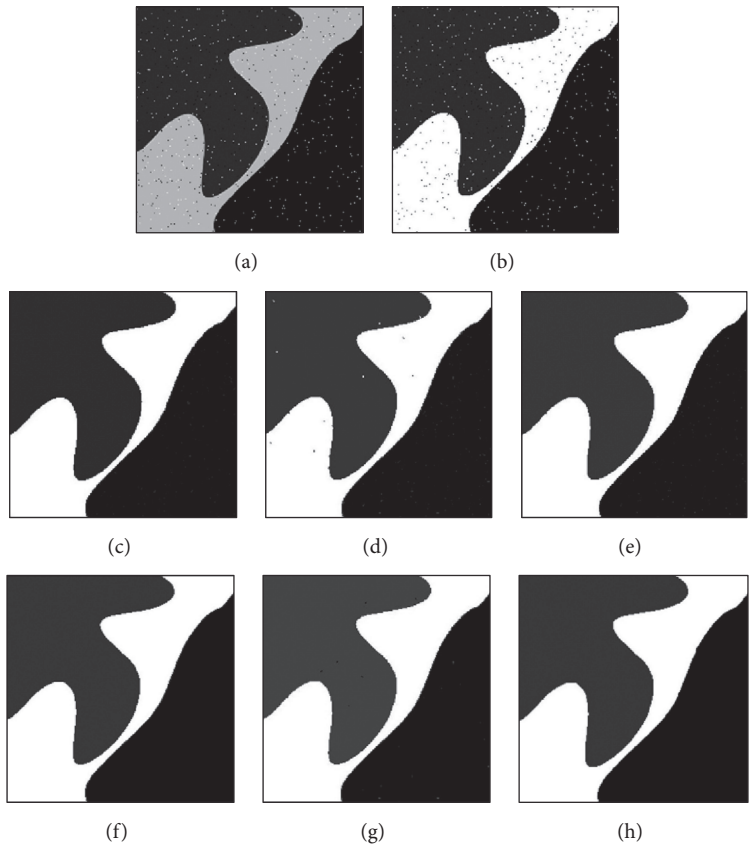

Figure 3: Segmentation results on synthetic image. (a) Original image corrupted by 15% of Salt and Pepper noise. (b) FCM result. (c) FCM_S result. (d) FCM_S1 result. (e) FCM_S2 result. (f) RFCMLGI result. (g) RFCMLGI_1 result. (h) RFCMLGI_2 result.

Table 1: Segmentation accuracies (SA, in %) of seven algorithms on synthetic image.

Algorithm	Gaussian			Salt and Pepper		
	9%	12%	15%	9%	12%	15%
FCM	91.24	88.73	86.55	97.89	97.76	97.50
FCM_S	98.23	97.94	97.58	98.21	98.18	98.10
RFCMLGI	**98.89**	**98.89**	**98.89**	**98.90**	**98.89**	**98.89**
FCM_S1	98.79	98.61	98.29	98.73	98.58	98.55
RFCMLGI_1	**98.76**	**98.66**	**98.44**	**98.91**	**98.86**	**98.86**
FCM_S2	98.93	98.72	98.33	98.89	98.79	98.77
RFCMLGI_2	**98.94**	**98.76**	**98.40**	**99.01**	**99.01**	**99.01**

is for the details presented on the coins. The segmentation results of all the algorithms are presented in Figures 4 and 5.

In Figures 4 and 5, we can see that our algorithms RFCMLGI, RFCMLGI_1, and RFCMLGI_2 performed better (to different extents) than the FCM_S, FCM_S1, and FCM_S2, respectively. Indeed, in case of Gaussian noise (Figure 4), we notice that the RFCMLGI has the best performance followed by the RFCMLGI_1. Thus, these two latter algorithms are the most favorable in the presence of Gaussian noise. The failure of the RFCMLGI_2 to deal correctly with noise is due to the median filter used inside the algorithm which is known by its weakness in removing Gaussian noise [21].

Under Salt and Pepper noise (Figure 5), we can order the algorithms according to their performances from the best to the worst as follows: RFCMLGI, RFCMLGI_2, FCM_S2, RFCMLGI_1, FCM_S1, FCM_S, and FCM. In addition to their best results, the RFCMLGI and RFCMLGI_2 performed very similarly, which means that they are both very convenient to handle Salt and Pepper noise. The RFCMLGI_1 could not achieve the best performance because of using the mean filter that is not recommended for Salt and Pepper noise.

In terms of detail preserving, we notice clearly (from Figures 4 and 5) that the RFCMLGI algorithm surpassed all the algorithms, where it found a good balance between noise elimination and detail preserving.

As has been concluded in the previous section, the RFCMLGI algorithm is the most convenient one when noise is a priori unknown.

3.2.3. Moon Image. To show the effect of our algorithms on images with mixed noise, we use the "*moon*" image corrupted at the same time by 20% of Gaussian and Salt and Pepper noise. In this experiment, C is fixed to 2. The segmentation results are shown in Figure 6.

From Figure 6, we note that the standard FCM has the worst performance. In contrast, the FCM_S, FCM_S1,

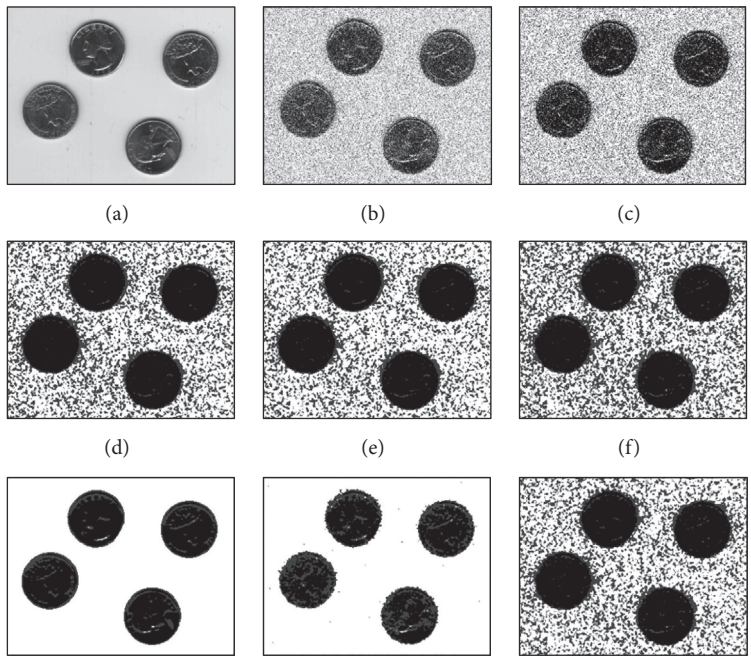

Figure 4: Segmentation result on *eight* image. (a) Originale image. (b) Image corrupted by 30% of Gaussian noise. (c) FCM result. (d) FCM_S result. (e) FCM_S1 result. (f) FCM_S2 result. (g) RFCMLGI result. (h) RFCMLGI_1 result. (i) RFCMLGI_2 result.

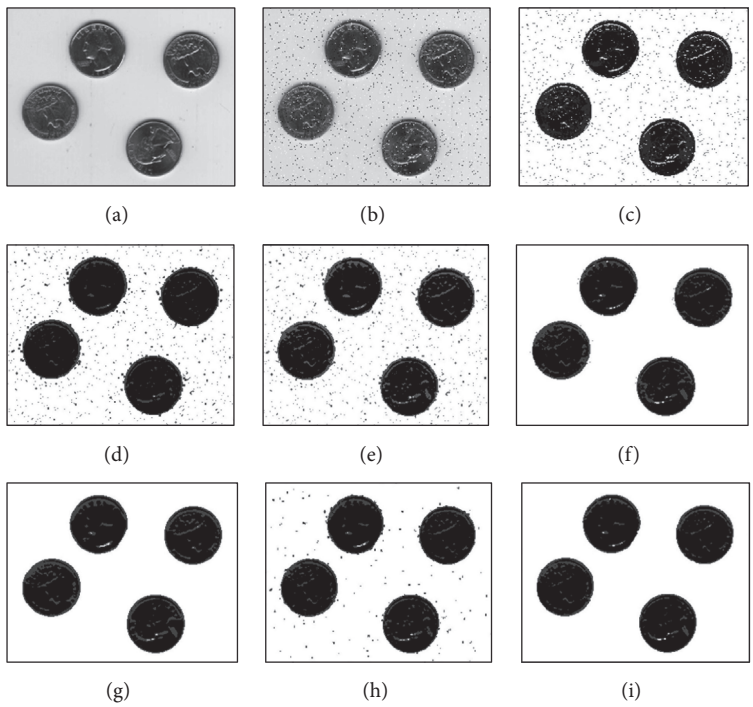

Figure 5: Segmentation result on *eight* image. (a) Original image. (b) Image corrupted by 30% of Salt and Pepper noise. (c) FCM result. (d) FCM_S result. (e) FCM_S1 result. (f) FCM_S2 result. (g) RFCMLGI result. (h) RFCMLGI_1 result. (i) RFCMLGI_2 result.

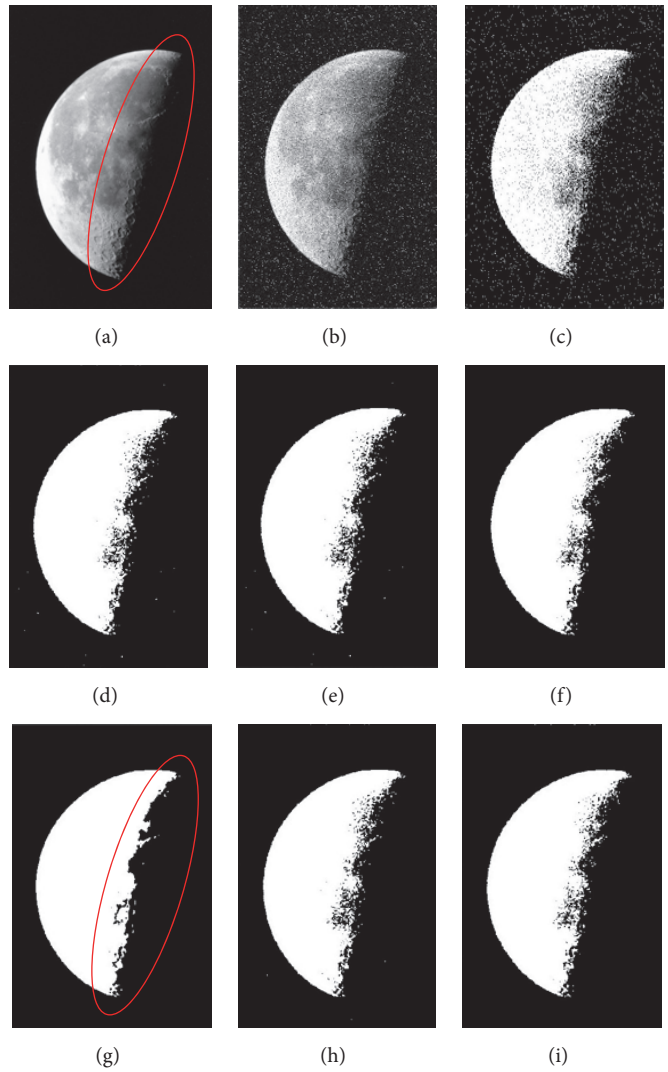

Figure 6: Segmentation result on *moon* image. (a) Original image. (b) Image corrupted by Gaussian and "Salt and Pepper" noise. (c) FCM result. (d) FCM_S result. (e) FCM_S1 result. (f) FCM_S2 result. (g) RFCMLGI result. (h) RFCMLGI_1 result. (i) RFCMLGI_2 result.

FCM_S2, RFCMLGI_1, and RFCMLGI_2 succeeded in handling noisy pixels and their performances are close to each other. However, the RFCMLGI algorithm made an exception where it outperformed all the algorithms in handling fuzzy pixels of the intersection region between the moon and the background (see regions circled in red), which proves its ability to retain fine details.

Globally, in the experimental results presented in this section we found that the proposed algorithms RFCMLGI, RFCMLGI_1, and RFCMLGI_2 performed better than the FCM_S, FCM_S1, and FCM_S2, respectively, and the RFCMLGI had the best performance. Even though in some cases RFCMLGI_1 and RFCMLGI_2 performed closely to the FCM_S1 and FCM_S2, they remain better because they are free of any parameter selection and they control the effect of the neighboring term automatically.

The standard FCM and its extensions FCM_S, FCM_S1, and FCM_S2 have the same time complexity which is O(HWC) [22], where H and W are the image dimensions and C is the number of clusters. The proposed algorithms RFCMLGI, RFCMLGI_1, and RFCMLGI_2 are similar to their antecedents FCM_S, FCM_S1, and FCM_S2, respectively, where the difference lies in the parameter that controls the tradeoff between noise elimination and detail preserving. Thus, the proposed algorithms have also the same time complexity O(HWC) with small variations caused by computing the new factor S.

4. Conclusion

In order to furnish a fuzzy clustering algorithm that is fully free of empirical parameters and noise type-independent, this work extended the FCM_S and its two variants to three algorithms based on a new factor that uses the local spatial and the gray level information to calculate the weight of the neighboring term. Generally, all the proposed algorithms

RFCMLGI, RFCMLGI_1, and RFCMLGI_2 proved their efficiency on synthetic and real images. More specifically, the RFCMLGI algorithm surpassed considerably the others where it showed its noise type-independence and its ability to retain fine details.

In spite of their fruitful results, the proposed algorithms need to be improved in the running times point of view, where computing the factor S makes them slower. This drawback will be the main issue of a future work.

Competing Interests

The authors declared that there is no conflict of interests regarding the publication of this paper.

References

[1] S. Fayech, N. Essoussi, and M. Limam, "Partitioning clustering algorithms for protein sequence data sets," *BioData Mining*, vol. 2, no. 1, article 3, 2009.

[2] D. Aneja and T. K. Rawat, "Fuzzy clustering algorithms for effective medical image segmentation," *International Journal of Intelligent Systems and Applications*, vol. 5, no. 11, pp. 55–61, 2013.

[3] M. Gupta, "Target detection by fuzzy gustafson-kessel algorithm," *International Journal of Image Processing (IJIP)*, vol. 7, no. 2, p. 203, 2013.

[4] J. C. Bezdek, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Springer, New York, NY, USA, 2005.

[5] M. S. Choudhry and R. Kapoor, "Performance analysis of fuzzy C-means clustering methods for MRI image segmentation," *Procedia Computer Science*, vol. 89, pp. 749–758, 2016.

[6] E. Abdel-Maksoud, M. Elmogy, and R. Al-Awadi, "Brain tumor segmentation based on a hybrid clustering technique," *Egyptian Informatics Journal*, vol. 16, no. 1, pp. 71–81, 2015.

[7] D. L. Pham, "Spatial models for fuzzy clustering," *Computer Vision and Image Understanding*, vol. 84, no. 2, pp. 285–297, 2002.

[8] M. N. Ahmed, S. M. Yamany, N. Mohamed, A. A. Farag, and T. Moriarty, "A modified fuzzy C-means algorithm for bias field estimation and segmentation of MRI data," *IEEE Transactions on Medical Imaging*, vol. 21, no. 3, pp. 193–199, 2002.

[9] L. Szilágyi, Z. Benyó, S. M. Szilágyi, and H. S. Adam, "MR brain image segmentation using an enhanced fuzzy c-means algorithm," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 1, pp. 724–726, September 2003.

[10] S. Chen and D. Zhang, "Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 4, pp. 1907–1916, 2004.

[11] A. Cherkaoui and H. Barrah, "Fast robust fuzzy clustering algorithm for grayscale image segmentation," in *Proceedings of the Xème Conférence Internationale: Conception et Production Intégrées (CPI '15)*, Tangier, Morocco, December 2015.

[12] W. Cai, S. Chen, and D. Zhang, "Fast and robust fuzzy *c*-means clustering algorithms incorporating local information for image segmentation," *Pattern Recognition*, vol. 40, no. 3, pp. 825–838, 2007.

[13] K.-S. Chuang, H.-L. Tzeng, S. Chen, J. Wu, and T.-J. Chen, "Fuzzy c-means clustering with spatial information for image segmentation," *Computerized Medical Imaging and Graphics*, vol. 30, no. 1, pp. 9–15, 2006.

[14] F. Zheng, C. Zhang, X. Zhang, and Y. Liu, "A fast anti-noise fuzzy C-means algorithm for image segmentation," in *Proceedings of the 20th IEEE International Conference on Image Processing (ICIP '13)*, pp. 2728–2732, Melbourne, Australia, September 2013.

[15] Z. Ji, J. Liu, G. Cao, Q. Sun, and Q. Chen, "Robust spatially constrained fuzzy c-means algorithm for brain MR image segmentation," *Pattern Recognition*, vol. 47, no. 7, pp. 2454–2466, 2014.

[16] S. Ghosh and S. K. Dubey, "Comparative analysis of K-means and fuzzy C-means algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 4, pp. 35–38, 2013.

[17] D. Lam and D. C. Wunsch, "Chapter 20-Clustering," in *Academic Press Library in Signal Processing*, P. S. R. Diniz, J. A. K. Suykens, R. Chellappa, and S. Theodoridis, Eds., vol. 1, pp. 1115–1149, 2014.

[18] L. Rokach and O. Maimon, "Clustering methods," in *Data Mining and Knowledge Discovery Handbook*, pp. 321–352, Springer, Berlin, Germany, 2005.

[19] D. Dubois, W. Ostasiewicz, and H. Prade, "Fuzzy sets: history and basic notions," in *Fundamental of Fuzzy Sets*, Kluwer Academic, Boston, Mass, USA, 2000.

[20] I. Ozkan and I. B. Turksen, "Upper and lower values for the level of fuzziness in FCM," *Information Sciences*, vol. 177, no. 23, pp. 5143–5152, 2007.

[21] A. B. Hamza and H. Krim, "Image denoising: a nonlinear robust statistical approach," *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 3045–3054, 2001.

[22] S. Krinidis and V. Chatzis, "A robust fuzzy local information C-means clustering algorithm," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1328–1337, 2010.

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2016, Article ID 3673267, 10 pages
http://dx.doi.org/10.1155/2016/3673267

Research Article

Fuzzy Constrained Probabilistic Inventory Models Depending on Trapezoidal Fuzzy Numbers

Mona F. El-Wakeel[1,2] and Kholood O. Al-yazidi[1]

[1] *Department of Statistics and Operations Researches, College of Science, King Saud University, P.O. Box 22452, Riyadh 11495, Saudi Arabia*
[2] *Higher Institute for Computers, Information and Management Technology, Tanta, Egypt*

Correspondence should be addressed to Mona F. El-Wakeel; melwakeel@ksu.edu.sa

Received 30 March 2016; Accepted 26 July 2016

Academic Editor: Gözde Ulutagay

We discussed two different cases of the probabilistic continuous review mixture shortage inventory model with varying and constrained expected order cost, when the lead time demand follows some different continuous distributions. The first case is when the total cost components are considered to be crisp values, and the other case is when the costs are considered as trapezoidal fuzzy number. Also, some special cases are deduced. To investigate the proposed model in the crisp case and the fuzzy case, illustrative numerical example is added. From the numerical results we will conclude that Uniform distribution is the best distribution to get the exact solutions, and the exact solutions for fuzzy models are considered more practical and close to the reality of life and get minimum expected total cost less than the crisp models.

1. Introduction

Inventory system is one of the most diversified fields of applied sciences that are widely used in a variety of areas including operations research, applied probability, computer sciences, management sciences, production system, and telecommunications. More than fifty years ago, the analysis of inventory system has appeared in the reference books and survey papers. Hadley and Whitin [1] are considered one of the first researchers who have discussed the analysis of inventory systems, where they displayed a method for the analysis of the mathematical model for inventory systems. Also, Balkhi and Benkherouf [2] have introduced production lot size inventory model in which products deteriorate at a constant rate and in which demand and production rates are allowed to vary with time. Inventory models may be either deterministic or probabilistic, since the demand of commodity may be deterministic or probabilistic, respectively. These cases were dealt with by Hadley and Whitin [1], Abuo-El-Ata et al. [3], and Vijayan and Kumaran [4].

Some managers allow the shortage in inventory systems; this shortage may be backorder case, lost sales case, and mixture shortage case. Many authors are dealing with inventory problems with various shortage cases where the cost components are considered as crisp values which does not depict the real inventory system fully. For example, constrained probabilistic inventory model with varying order and shortage costs using Lagrangian method has been investigated by Fergany [5]. In addition, constrained probabilistic inventory model with continuous distributions and varying holding cost was discussed by Fergany and El-Saadani [6]. In 2006, several models of continuous distributions for constrained probabilistic lost sales inventory models with varying order cost under holding cost constraint using Lagrangian method by Fergany and El-Wakeel [7, 8] were discussed. Recently, El-Wakeel [9] deduced constrained backorders inventory system with varying order cost under holding cost constraint: lead time demand uniformly distributed using Lagrangian method. Also, El-Wakeel and Fergany [10] deduced constrained probabilistic continuous review inventory system with mixture shortage and stochastic lead time demand.

Sometimes, the cost components are considered as fuzzy values, because, in real life, the various physical or chemical

characteristics may cause an effect on the cost components and then precise values of cost characteristics become difficult to measure as the exact amount of order, holding, and especially shortage cost. Thus, in controlling the inventory system it may allow some flexibility in the cost parameter values in order to treat the uncertainties which always fit the real situations. Since we want to satisfy our requirements for such contradictions, the fuzzy set theory meets these requirements to some extent. In 1965, Zadeh [11] first introduced the fuzzy set theory which studied the intention to accommodate uncertainty in the nonstochastic sense rather than the presence of random variables. Syed and Aziz [12] have examined the fuzzy inventory model without shortages using signed distance method. Kazemi et al. [13] have treated the inventory model with backorders with fuzzy parameters and decision variables. Gawdt [14] presented a mixture continuous review inventory model under varying holding cost constraint when the lead time demand follows Gamma distribution, where the costs were fuzzified as the trapezoidal fuzzy numbers. The continuous review inventory model with mixture shortage under constraint involving crashing cost based on probabilistic triangular fuzzy numbers by Fergany and Gawdt [15] was discussed. A probabilistic periodic review inventory model using Lagrange technique and fuzzy adaptive particle swarm optimization was presented by Fergany et al. [16]. Fuzzy inventory model for deteriorating items with time dependent demand and partial backlogging is established by Kumar and Rajput [17]. Indrajitsingha et al. [18] give fuzzy inventory model with shortages under fully backlogged using signed distance method. Recently, Patel et al. [19] introduced the continuous review inventory model under fuzzy environment without backorder for deteriorating items.

As we found earlier, many authors have studied the inventory models with different assumptions and conditions. These assumptions and conditions are represented in constraints and costs (constant or varying). Therefore, due to the importance of the inventory models we shall propose and study, in this paper, the mixture shortage inventory model with varying order cost under expected order cost constraint and the lead time demand follow Exponential, Laplace, and Uniform distributions. Our goal of studying the inventory models is to minimize the total cost. The order quantity and the reorder point are the policy variables for this model, which minimize the expected annual total cost. We evaluated the optimal order quantity and the reorder point in two cases: first case is when the cost components are considered as crisp values, and the second case is when the cost components are fuzzified as a trapezoidal fuzzy numbers, which is called the fuzzy case. Finally this work is illustrated by numerical example and we will make comparisons of all results and obtain conclusions.

2. Model Development

To develop any model of inventory models we need to put some notations and assumptions represented in Notations section.

2.1. Assumptions

(1) Consider that continuous review inventory model under order cost constraint and shortages are allowed.

(2) Demand is a continuous random variable with known probability.

(3) The lead time is constant and follows the known distributions.

(4) γ is a fraction of unsatisfied demand that will be backordered while the remaining fraction $(1-\gamma)$ is completely lost, where $0 \le \gamma \le 1$.

(5) New order with size (Q) is placed when the inventory level drops to a certain level, called the reorder point (r); assume that the system repeats itself in the sense that the inventory position varies between r and $r+Q$ during each cycle.

3. Model (I): The Mixture Shortage Model Where the Cost Components Are Considered as Crisp Values

In this section, we consider that the continuous review inventory model with shortage is allowed. Some customers are willing to wait for the new replenishment and the others have no patience; this case is called mixture shortage or partial backorders.

The expected annual total cost consisted of the sum of three components:

$$\begin{aligned} E(\text{Total Cost}) &= E(\text{order Cost}) + E(\text{Holding Cost}) \\ &\quad + E(\text{Shortage Cost}), \\ E(\text{TC}(Q,r)) &= E(\text{OC}) + E(\text{HC}) + E(\text{SC}), \end{aligned} \tag{1}$$

where

$$E(\text{SC}) = E(\text{BC}) + E(\text{LC}) \tag{2}$$

and we assume the varying order cost function, where the order cost is a decreasing function of the order quantity Q. Then, the expected order cost is given by

$$\begin{aligned} E(\text{OC}) &= c_o(Q)\frac{\overline{D}}{Q} = c_o Q^{-\beta}\frac{\overline{D}}{Q} = c_o \overline{D} Q^{-\beta-1}, \\ E(\text{HC}) &= c_h \overline{H} = c_h\left[\frac{Q}{2} + r - E(x) + (1-\gamma)\overline{S}(r)\right], \\ E(\text{BC}) &= \frac{c_b \gamma \overline{D}}{Q}\overline{S}(r), \\ E(\text{LC}) &= \frac{c_l(1-\gamma)\overline{D}}{Q}\overline{S}(r). \end{aligned} \tag{3}$$

Our objective is to minimize the expected total costs $[\min E(\text{TC}(Q,r))]$ with varying order cost under the expected order cost constraint which needs to find the

optimal values of order quantity Q and reorder point r. To solve this primal function, let us write it as follows:

$$\begin{aligned} E(\mathrm{TC}(Q,r)) &= c_o\overline{D}Q^{-\beta-1} \\ &\quad + c_h\left[\frac{Q}{2}+r-E(x)+(1-\gamma)\,\overline{S}(r)\right] \\ &\quad + \frac{c_b\gamma\overline{D}}{Q}\overline{S}(r)+\frac{c_l(1-\gamma)\,\overline{D}}{Q}\overline{S}(r) \\ &= c_o\overline{D}Q^{-\beta-1}+c_h\left(\frac{Q}{2}+r-E(x)\right) \\ &\quad + \frac{c_b\gamma\overline{D}}{Q}\overline{S}(r) \\ &\quad + \left(c_h+\frac{c_l\overline{D}}{Q}\right)(1-\gamma)\,\overline{S}(r) \end{aligned} \tag{4}$$

$$\text{Subject to:}\quad c_o\overline{D}Q^{-\beta-1}\le K. \tag{5}$$

We use the Lagrange multiplier technique to get the optimal values Q^* and r^* which minimize (4) under constraint (5) as follows:

$$\begin{aligned} G(Q,r,\lambda) &= c_o\overline{D}Q^{-\beta-1}+c_h\left(\frac{Q}{2}+r-E(x)\right) \\ &\quad + \frac{c_b\gamma\overline{D}}{Q}\overline{S}(r)+\left(c_h+\frac{c_l\overline{D}}{Q}\right)(1-\gamma)\,\overline{S}(r) \\ &\quad + \lambda\left(c_o\overline{D}Q^{-\beta-1}-K\right). \end{aligned} \tag{6}$$

Putting each of the corresponding first partial derivatives of (6) equal to zero at $Q=Q^*$ and $r=r^*$, respectively, we get

$$\begin{aligned} c_hQ^{*2}+B\overline{D}Q^{*-\beta}-2A\overline{S}(r^*)&=0, \\ R(r^*)&=\frac{c_hQ^*}{c_h(1-\gamma)Q^*+A}, \end{aligned} \tag{7}$$

where

$$\begin{aligned} A&=\overline{D}\left[c_b\gamma+c_l(1-\gamma)\right], \\ B&=2c_o(-\beta-1)\left[1+\lambda\right]. \end{aligned} \tag{8}$$

Clearly, it is difficult to find an exact solution of Q^* and r^* of (7), so we can suppose that the lead time demand follows some distributions.

3.1. Lead Time Demand Follows Exponential Distribution. Supposing that the lead time demand follows the Exponential distribution with parameters ν, then its probability density function is given by

$$\begin{aligned} f(x)&=\nu e^{-\nu x};\quad x\ge 0,\ \nu>0 \\ \text{with } E(x)&=\frac{1}{\nu}, \\ R(r)&=e^{-\nu r}, \\ \overline{S}(r)&=\frac{1}{\nu}e^{-\nu r}. \end{aligned} \tag{9}$$

The optimal order quantity and the optimal reorder level which minimize the expected relevant annual total cost can be obtained by substituting (9) into (7). Solving them simultaneously we get

$$\begin{aligned} &\nu c_h^2(1-\gamma)Q^{*3}+\nu c_hAQ^{*2}-2c_hAQ^* \\ &\quad +\nu c_h(1-\gamma)B\overline{D}Q^{*1-\beta}+\nu AB\overline{D}Q^{*-\beta}=0, \\ &r^*=-\frac{1}{\nu}\ln\left[\frac{c_hQ^*}{c_h(1-\gamma)Q^*+A}\right], \end{aligned} \tag{10}$$

which give exact solutions for model (I).

3.2. Lead Time Demand Follows Laplace Distribution. If the lead time demand follows the Laplace distribution with parameters μ,θ, the probability density function will be

$$\begin{aligned} f(x)&=\frac{1}{2\theta}e^{-|x-\mu|/\theta};\quad -\infty<x<\infty,\ \theta>0 \\ \text{with } E(x)&=\mu, \\ R(r)&=\frac{1}{2}e^{-((r-\mu)/\theta)}, \\ \overline{S}(r)&=\frac{\theta}{2}e^{-((r-\mu)/\theta)}. \end{aligned} \tag{11}$$

The optimal order quantity and the optimal reorder level which minimize the expected relevant annual total cost can be obtained by substituting (11) into (7), and, solving them simultaneously, we obtain

$$\begin{aligned} &c_h^2(1-\gamma)Q^{*3}+c_hAQ^{*2}-2c_h\theta AQ^* \\ &\quad +c_h(1-\gamma)B\overline{D}Q^{*1-\beta}+AB\overline{D}Q^{*-\beta}=0, \\ &r^*=\mu-\theta\ln\left[\frac{2c_hQ^*}{c_h(1-\gamma)Q^*+A}\right], \end{aligned} \tag{12}$$

which give exact solutions for model (I).

3.3. Lead Time Demand Follows Uniform Distribution. Similarly, suppose that the lead time demand follows the Uniform distribution over the range from 0 to b, that is, $x \sim \mathrm{Uniform}(0,b)$; then its probability density function is given by

$$\begin{aligned} f(x)&=\frac{1}{b};\quad 0\le x\le b \\ \text{with } E(x)&=\frac{b}{2}, \\ R(r)&=1-\frac{r}{b}, \\ \overline{S}(r)&=\frac{1}{2b}(r-b)^2. \end{aligned} \tag{13}$$

The optimal order quantity and the optimal reorder level which minimize the expected relevant annual total cost

can be obtained by substituting (13) into (7). Solving them simultaneously, we find

$$\begin{aligned} & c_h^3 (1-\gamma)^2 Q^{*4} + 2c_h^2 (1-\gamma) AQ^{*3} \\ & \quad + c_h A [A - bc_h] Q^{*2} + c_h^2 (1-\gamma)^2 B\overline{D} Q^{*2-\beta} \\ & \quad + 2c_h (1-\gamma) AB\overline{D} Q^{*1-\beta} + A^2 B\overline{D} Q^{*-\beta} = 0, \\ & r^* = b\left[1 - \frac{c_h Q^*}{c_h (1-\gamma) Q^* + A}\right], \end{aligned} \tag{14}$$

which give exact solutions for model (I).

Thus, the exact solution for constrained continuous review inventory model with mixture shortage and varying order cost can obtained by solving previous equations for each distribution separately at different values of β and varying λ until the smallest positive value is found such that the constraint holds.

4. Model (I_f): The Mixture Shortage Model Where the Cost Components Are Considered as Fuzzy Numbers

Consider continuous review inventory model similar to model (I), but assuming that the cost components c_o, c_h, c_b, and c_l are all fuzzy numbers, to control various uncertainties from various physical or chemical characteristics where there may be an effect on the cost components.

We represent these costs by trapezoidal fuzzy numbers as given below:

$$\begin{aligned} \widetilde{c}_o &= (c_o - \delta_1, c_o - \delta_2, c_o + \delta_3, c_o + \delta_4), \\ \widetilde{c}_h &= (c_h - \delta_5, c_h - \delta_6, c_h + \delta_7, c_h + \delta_8), \\ \widetilde{c}_b &= (c_b - \theta_1, c_b - \theta_2, c_b + \theta_3, c_b + \theta_4), \\ \widetilde{c}_l &= (c_l - \theta_5, c_l - \theta_6, c_l + \theta_7, c_l + \theta_8), \end{aligned} \tag{15}$$

where δ_i and θ_i, $i = 1, 2, \ldots, 8$ are arbitrary positive numbers and should satisfy the following constraints:

$$\begin{aligned} & c_o > \delta_1 > \delta_2, \\ & \delta_3 < \delta_4, \\ & c_h > \delta_5 > \delta_6, \\ & \delta_7 < \delta_8. \end{aligned} \tag{16}$$

Similarly,

$$\begin{aligned} & C_b > \theta_1 > \theta_2, \\ & \theta_3 < \theta_4, \\ & c_l > \theta_5 > \theta_6, \\ & \theta_7 < \theta_8. \end{aligned} \tag{17}$$

We can represent the order cost as a trapezoidal fuzzy number as shown in Figure 1 and similarly for the remaining costs.

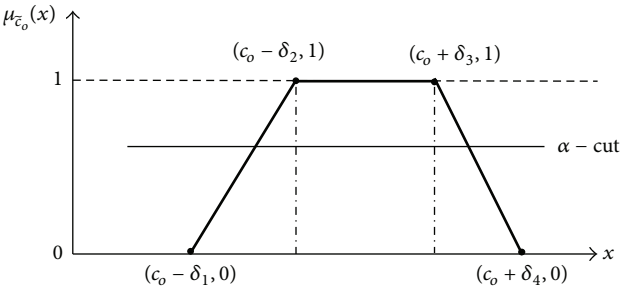

Figure 1: Order cost as a trapezoidal fuzzy number.

Note that the membership function of $\widetilde{c}_o$ is 1 at points $c_o - \delta_2$ and $c_o + \delta_3$, decreases as the point deviates from $c_o - \delta_2$ and $c_o + \delta_3$, and reaches zero at the endpoints $c_o - \delta_1$ and $c_o + \delta_4$.

The left and right limits of α – cut of c_o, c_h, c_b, and c_l are given as follows:

$$\begin{aligned} \widetilde{c}_{ov}(\alpha) &= c_o - \delta_1 + (\delta_1 - \delta_2)\alpha, \\ \widetilde{c}_{ou}(\alpha) &= c_o + \delta_4 - (\delta_4 - \delta_3)\alpha, \\ \widetilde{c}_{hv}(\alpha) &= c_h - \delta_5 + (\delta_5 - \delta_6)\alpha, \\ \widetilde{c}_{hu}(\alpha) &= c_h + \delta_8 - (\delta_8 - \delta_7)\alpha, \\ \widetilde{c}_{bv}(\alpha) &= c_b - \theta_1 + (\theta_1 - \theta_2)\alpha, \\ \widetilde{c}_{bu}(\alpha) &= c_b + \theta_4 - (\theta_4 - \theta_3)\alpha, \\ \widetilde{c}_{lv}(\alpha) &= c_l - \theta_5 + (\theta_5 - \theta_6)\alpha, \\ \widetilde{c}_{lu}(\alpha) &= c_l + \theta_8 - (\theta_8 - \theta_7)\alpha. \end{aligned} \tag{18}$$

The expected annual total cost for this case under the expected order cost constraint and when all cost components are fuzzy can be expressed as follows:

$$\begin{aligned} \widetilde{E}(\widetilde{c}_o, \widetilde{c}_h, \widetilde{c}_b, \widetilde{c}_l) &= \widetilde{c}_o \overline{D} Q^{-\beta-1} \\ &\quad + \widetilde{c}_h \left[\frac{Q}{2} + r - E(x) + (1-\gamma)\overline{S}(r)\right] \\ &\quad + \frac{\widetilde{c}_b \gamma \overline{D}}{Q}\overline{S}(r) + \frac{\widetilde{c}_l (1-\gamma)\overline{D}}{Q}\overline{S}(r) \\ &= \widetilde{c}_o \overline{D} Q^{-\beta-1} + \widetilde{c}_h \left(\frac{Q}{2} + r - E(x)\right) \\ &\quad + \frac{\widetilde{c}_b \gamma \overline{D}}{Q}\overline{S}(r) \\ &\quad + \left(\widetilde{c}_h + \frac{\widetilde{c}_l \overline{D}}{Q}\right)(1-\gamma)\overline{S}(r) \end{aligned} \tag{19}$$

$$\text{Subject to:} \quad \widetilde{c}_o \overline{D} Q^{-\beta-1} \le K. \tag{20}$$

We use the Lagrange multiplier technique to find the optimal values Q^* and r^* which minimize (19) under constraint (20) as follows:

$$\begin{aligned}\widetilde{G}\left(\widetilde{c}_o,\widetilde{c}_h,\widetilde{c}_b,\widetilde{c}_l\right) &= \widetilde{c}_o\overline{D}Q^{-\beta-1} + \widetilde{c}_h\left(\frac{Q}{2}+r-E\left(x\right)\right)\\ &+ \frac{\widetilde{c}_b\gamma\overline{D}}{Q}\overline{S}\left(r\right)\\ &+ \left(\widetilde{c}_h+\frac{\widetilde{c}_l\overline{D}}{Q}\right)\left(1-\gamma\right)\overline{S}\left(r\right)\\ &+ \lambda\left(\widetilde{c}_o\overline{D}Q^{-\beta-1}-K\right).\end{aligned} \tag{21}$$

We can obtain the form of left and right α – cut of the fuzzified cost function (21), respectively, as follows:

$$\begin{aligned}\widetilde{G}\left(\widetilde{c}_o,\widetilde{c}_h,\widetilde{c}_b,\widetilde{c}_l\right)_v\left(\alpha\right) &= \widetilde{c}_{ov}\overline{D}Q^{-\beta-1}\\ &+ \widetilde{c}_{hv}\left(\frac{Q}{2}+r-E\left(x\right)\right)\\ &+ \frac{\widetilde{c}_{bv}\gamma\overline{D}}{Q}\overline{S}\left(r\right)\\ &+ \left(\widetilde{c}_{hv}+\frac{\widetilde{c}_{lv}\overline{D}}{Q}\right)\left(1-\gamma\right)\overline{S}\left(r\right)\\ &+ \lambda\left(\widetilde{c}_{ov}\overline{D}Q^{-\beta-1}-K\right),\\ \widetilde{G}\left(\widetilde{c}_o,\widetilde{c}_h,\widetilde{c}_b,\widetilde{c}_l\right)_u\left(\alpha\right) &= \widetilde{c}_{ou}\overline{D}Q^{-\beta-1}\\ &+ \widetilde{c}_{hu}\left(\frac{Q}{2}+r-E\left(x\right)\right)\\ &+ \frac{\widetilde{c}_{bu}\gamma\overline{D}}{Q}\overline{S}\left(r\right)\\ &+ \left(\widetilde{c}_{hu}+\frac{\widetilde{c}_{lu}\overline{D}}{Q}\right)\left(1-\gamma\right)\overline{S}\left(r\right)\\ &+ \lambda\left(\widetilde{c}_{ou}\overline{D}Q^{-\beta-1}-K\right).\end{aligned} \tag{22}$$

Since $\widetilde{G}_v(\alpha)$ and $\widetilde{G}_u(\alpha)$ exist and are integrable for $\alpha \in [0,1]$, as in Yao and Wu [20], we have

$$d\left(\widetilde{G},\widetilde{0}\right) = \frac{1}{2}\int_0^1\left(\widetilde{G}_v\left(\alpha\right)+\widetilde{G}_u\left(\alpha\right)\right)d\alpha. \tag{23}$$

We get the defuzzified value of $\widetilde{G}(\widetilde{c}_o,\widetilde{c}_h,\widetilde{c}_b,\widetilde{c}_l)(\alpha)$ by using (23) for (22) as follows:

$$\begin{aligned}d\left(\widetilde{G},\widetilde{0}\right) &= A_1\overline{D}Q^{-\beta-1} + A_2\left(\frac{Q}{2}+r-E\left(x\right)\right)\\ &+ \frac{A_3\gamma\overline{D}}{Q}\overline{S}\left(r\right)\\ &+ \left(A_2+\frac{A_4\overline{D}}{Q}\right)\left(1-\gamma\right)\overline{S}\left(r\right)\\ &+ \lambda\left(A_1\overline{D}Q^{-\beta-1}-K\right),\end{aligned} \tag{24}$$

where

$$\begin{aligned}A_1 &= \frac{\left(4c_o-\delta_1-\delta_2+\delta_3+\delta_4\right)}{4},\\ A_2 &= \frac{\left(4c_h-\delta_5-\delta_6+\delta_7+\delta_8\right)}{4},\\ A_3 &= \frac{\left(4c_b-\theta_1-\theta_2+\theta_3+\theta_4\right)}{4},\\ A_4 &= \frac{\left(4c_l-\theta_5-\theta_6+\theta_7+\theta_8\right)}{4}.\end{aligned} \tag{25}$$

Similarly, as in model (I), to get the optimal values Q^* and r^* put each of the corresponding first partial derivatives of (24) equal to zero at $Q = Q^*$ and $r = r^*$, respectively; we obtain

$$\begin{aligned}&\left(2A_1\left(-\beta-1\right)\overline{D}Q^{*-\beta}\right)\left(1+\lambda\right)+A_2Q^{*2}\\ &\quad-2A_3\gamma\overline{DS}\left(r\right)-2A_4\left(1-\gamma\right)\overline{DS}\left(r\right)=0\end{aligned} \tag{26}$$

and the probability of the shortage is

$$R\left(r^*\right) = \frac{A_2Q^*}{A_2\left(1-\gamma\right)Q^*+A_3\gamma\overline{D}+A_4\left(1-\gamma\right)\overline{D}}. \tag{27}$$

Clearly, there is no closed form solution of (26) and (27). We can solve these equations by using the same manner as in model (I).

5. Special Cases

(1) Letting $\gamma = 0$, $\beta = 0$ and $K \to \infty \Rightarrow C_o(Q) = c_o$ and $\lambda = 0$, thus $A = c_l\overline{D}$, $B = -2c_o$ and hence (7) reduces to

$$\begin{aligned}Q^* &= \sqrt{\frac{2\overline{D}\left(c_o+c_l\overline{s}\left(r^*\right)\right)}{c_h}},\\ R\left(r^*\right) &= \frac{c_hQ^*}{c_hQ^*+c_l\overline{D}}.\end{aligned} \tag{28}$$

This is an unconstrained lost sales continuous review inventory model with constant units of costs, which are the same results as in Hadley and Whitin [1].

(2) Letting $\gamma = 1$, $\beta = 0$ and $K \to \infty \Rightarrow C_o(Q) = c_o$ and $\lambda = 0$, thus $A = c_b\overline{D}$, $B = -2c_o$; thus (7) reduces to

$$\begin{aligned}Q^* &= \sqrt{\frac{2\overline{D}\left(c_o+c_b\overline{s}\left(r^*\right)\right)}{c_h}},\\ R\left(r^*\right) &= \frac{c_hQ^*}{c_b\overline{D}}.\end{aligned} \tag{29}$$

This is an unconstrained backorders continuous review inventory model with constant units of costs, which are the same results as in Hadley and Whitin [1].

(i) Equations (10) give unconstrained backorders continuous review of inventory model with constant units of cost and the lead time demand follows the Exponential distribution, which are the same results as in Hillier and Lieberman [21].

(ii) Equations (12) give unconstrained backorders continuous review inventory model with constant units of cost and the lead time demand follows the Laplace distribution, which agree with results of Nahmias [22].

(iii) Equations (14) give unconstrained backorders continuous review inventory model with constant units of cost and the lead time demand follows the Uniform distribution, which are the same results as in Fabrycky and Banks [23].

6. Numerical Example

Consider an inventory system with the following data:

$\overline{D}$ = 1050 units per year,

c_o = 70 SR per unit ordered,

c_h = 25 SR per unit per year,

c_b = 7 SR per unit backorder,

c_l = 15 SR per unit lost,

the backorder fraction has the values $\gamma = 0.1$, $\gamma = 0.3$, and $\gamma = 0.7$,

let K = 140 SR,

and take

$$\delta_1 = 60,$$
$$\delta_2 = 48,$$
$$\delta_3 = 10,$$
$$\delta_4 = 50,$$
$$\delta_5 = 19,$$
$$\delta_6 = 10,$$
$$\delta_7 = 1,$$
$$\delta_8 = 2,$$
$$\theta_1 = 6,$$
$$\theta_2 = 4,$$
$$\theta_3 = 2,$$
$$\theta_4 = 4,$$
$$\theta_5 = 12,$$
$$\theta_6 = 7,$$

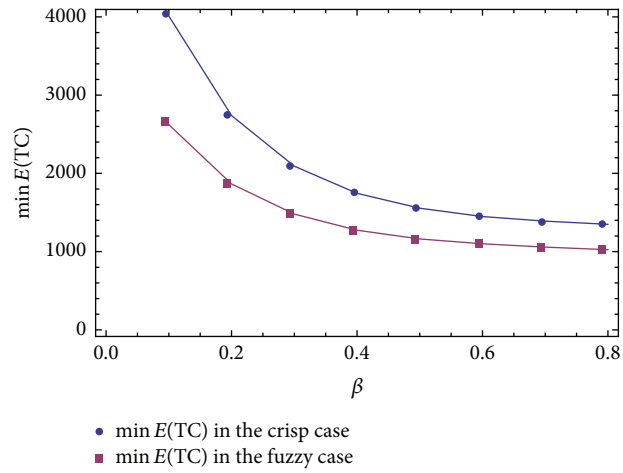

Figure 2: The comparison between the crisp and fuzzy cases for Exponential at $\gamma = 0.7$.

$$\theta_7 = 1,$$
$$\theta_8 = 2. \tag{30}$$

Determine Q^* and r^* for both cases of the previous model, when the lead time demand has the following distributions:

(i) Exponential distribution with $\nu = 0.077$ units.

(ii) Laplace distribution with $\mu = 13$ and $\theta = 10$ units.

(iii) Uniform distribution with $b = 26$ units.

Depending on the above data, we can obtain all results by solving the previous deduced equations at different values of β, λ, and γ as shown in the Tables 1, 2, and 3 which give the optimal values of Q^* and r^* that minimize the expected total cost, when the lead time demand follows Exponential, Laplace, and Uniform distribution, respectively, for model (I) and model (I_f).

From Table 1 we have that

at $\gamma = 0.1$, we will make backorders by 10% of new orders quantity;

at $\gamma = 0.3$, we will make backorders by 30% of new orders quantity;

at $\gamma = 0.7$, we will make backorders by 70% of new orders quantity.

After comparison of the crisp case and fuzzy case for Exponential distribution, we can deduce that the least min E(TC) was obtained at $\gamma = 0.7$. We can draw the minimum expected total cost for model (I) and model (I_f) against β for the Exponential distribution at $\gamma = 0.7$ as shown in Figure 2.

From Table 2 we have that

at $\gamma = 0.1$, we will make backorders by 10% of new orders quantity;

at $\gamma = 0.3$, we will make backorders by 30% of new orders quantity;

Table 1: The exact solutions and min E(TC) for model (I) and model (I_f) at Exponential distribution.

γ	β	Crisp case			Fuzzy case		
		Q^*	r^*	min E(TC)	Q^*	r^*	min E(TC)
0.1	0.1	297.092	13.8608	4199.84	250.412	15.426	2741.44
	0.2	184.893	18.4055	2910.93	158.061	20.016	1972.09
	0.3	123.76	22.6467	2252.78	107.107	24.237	1578.8
	0.4	87.6955	26.5107	1898.63	76.6735	28.054	1367.98
	0.5	65.1132	29.9807	1703	57.4583	31.4585	1253.06
	0.6	50.1308	33.1065	1593.97	46.7207	33.9499	1189.84
	0.7	41.8943	35.2866	1533.94	43.0554	34.9436	1146.05
	0.8	39.0054	36.1611	1491.79	39.9907	35.846	1112.46
0.3	0.1	297.079	11.8026	4148.23	250.456	13.6144	2708.31
	0.2	184.905	16.4977	2863.37	158.08	18.3589	1941.59
	0.3	123.759	20.8394	2207.59	107.132	22.6783	1550.15
	0.4	87.7063	24.768	1855.18	76.7397	26.5535	1340.67
	0.5	65.1619	28.2749	1660.8	57.4816	30.0074	1226.34
	0.6	50.1482	31.4365	1552.36	46.8468	32.4964	1163.56
	0.7	41.9983	33.6078	1492.73	43.1748	33.498	1119.93
	0.8	39.1053	34.4876	1450.74	40.1052	34.4067	1086.48
0.7	0.1	297.118	6.33535	4012.01	250.446	8.99977	2622.85
	0.2	184.851	11.5619	2739.35	158.081	14.2364	1865.33
	0.3	123.742	16.2344	2092.27	107.118	18.864	1479.48
	0.4	87.6963	20.3768	1745.29	76.7206	22.9372	1273.64
	0.5	65.1605	24.0241	1554.52	57.4204	26.5318	1161.7
	0.6	50.2168	27.266	1448.65	47.2184	28.9822	1100.36
	0.7	42.327	29.4105	1390.22	43.5262	30.0069	1057.21
	0.8	39.4206	30.3066	1348.72	40.4411	30.9342	1024.17

at $\gamma = 0.7$, we will make backorders by 70% of new orders quantity.

After comparison of the crisp case and fuzzy case for Laplace distribution, we can deduce that the least min E(TC) was obtained at $\gamma = 0.7$. We can draw the minimum expected total cost for model (I) and model (I_f) against β for the Laplace distribution at $\gamma = 0.7$ as shown in Figure 3.

From Table 3 we have that

at $\gamma = 0.1$, we will make backorders by 10% of new orders quantity;

at $\gamma = 0.3$, we will make backorders by 30% of new orders quantity;

at $\gamma = 0.7$, we will make backorders by 70% of new orders quantity.

After comparison of the crisp case and fuzzy case for Uniform distribution, we can deduce that the least min E(TC) was obtained at $\gamma = 0.7$. We can draw the minimum expected total cost for model (I) and model (I_f) against β for the Uniform distribution at $\gamma = 0.7$ as shown in Figure 4.

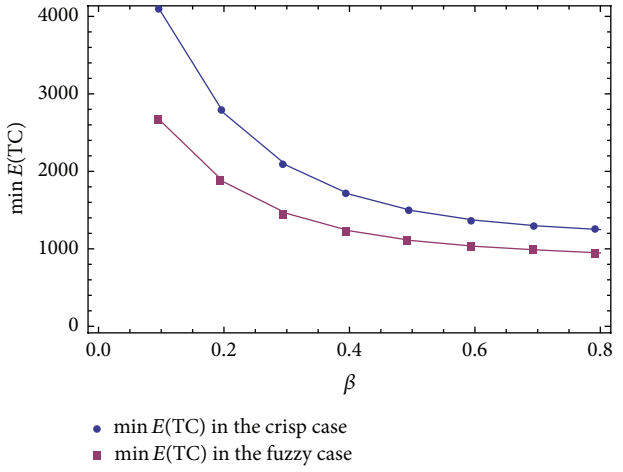

Figure 3: The comparison between the crisp and fuzzy cases for Laplace at $\gamma = 0.7$.

7. Conclusion

In this study we discussed two cases for mixture shortage inventory model under varying order cost constraint when lead time demand follows Exponential, Laplace, and Uniform distributions. We have evaluated the exact solutions of Q^* and r^* for each value of β and λ^* which yields our expected order cost constraint and then obtain the minimum expected total cost by using Lagrangian multiplier technique.

By comparing between the minimum expected total cost for model (I) and model (I_f) at each distribution, we can

Table 2: The exact solutions and min E(TC) for model (I) and model (I_f) at Laplace distribution.

γ	β	Crisp case			Fuzzy case		
		Q^*	r^*	min E(TC)	Q^*	r^*	min E(TC)
0.1	0.1	297.123	16.7406	4197.53	250.409	17.9466	2732.79
	0.2	184.844	20.2428	2881.62	158.099	21.4789	1944.2
	0.3	123.72	23.5092	2199.23	107.093	24.7322	1532.59
	0.4	87.7213	26.4792	1823.44	76.7474	27.6615	1305.96
	0.5	65.081	29.1581	1607.46	57.4314	30.2959	1176.14
	0.6	50.1314	31.5604	1480.65	44.6116	32.6421	1100.83
	0.7	39.9135	33.6954	1405.74	40.0084	33.6658	1052.46
	0.8	35.8352	34.715	1357.91	36.8761	34.4363	1014.82
0.3	0.1	297.09	15.1563	4157.53	250.449	16.5519	2707.33
	0.2	184.858	18.7738	2845.06	158.053	20.2063	1920.28
	0.3	123.74	22.1162	2164.62	107.069	23.536	1510.28
	0.4	87.6956	25.141	1789.72	76.6717	26.5228	1284.38
	0.5	65.0979	27.8493	1574.89	57.4223	29.1839	1155.52
	0.6	50.2117	30.2628	1448.85	44.5733	31.5604	1080.66
	0.7	39.8244	32.4508	1374.04	40.0876	32.5659	1032.46
	0.8	35.9	33.4388	1326.43	36.9506	33.3419	994.931
0.7	0.1	297.084	10.9477	4052.24	250.405	12.9997	2641.24
	0.2	184.853	14.9711	2749.93	158.116	17.0285	1862.01
	0.3	123.776	18.5665	2076.28	107.095	20.5958	1456.08
	0.4	87.7171	21.7564	1705.32	76.744	23.7273	1233.15
	0.5	65.084	24.5783	1492.99	57.4997	26.4847	1106.02
	0.6	50.1989	27.0667	1368.85	44.5736	28.9432	1032.24
	0.7	39.9021	29.2867	1295.45	40.3191	29.9172	984.487
	0.8	36.1029	30.2592	1248.3	37.1679	30.7094	947.275

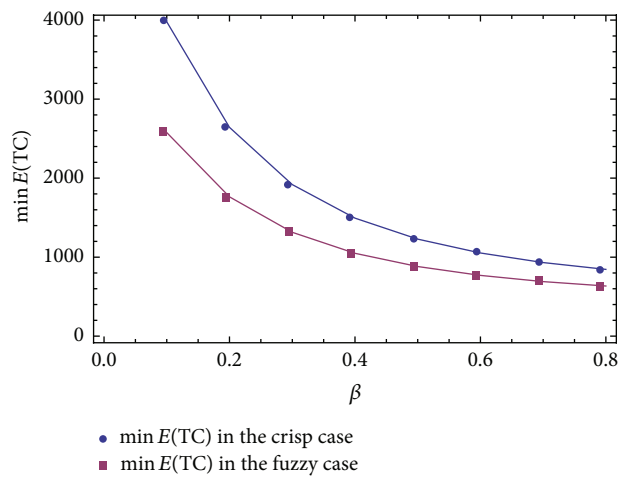

Figure 4: The comparison between the crisp and fuzzy cases for Uniform at $\gamma = 0.7$.

deduce that the least min E(TC) was obtained when the lead time demand follows Uniform distribution and equals 844.584 SR with order quantity $Q^* = 32.4596$ and reorder point $r^* = 23.9138$ for model (I), while the minimum expected annual total cost for model (I_f) is 634.709 SR with order quantity $Q^* = 29.3328$ and reorder point $r^* = 24.2447$ as shown in Table 3. This means that we can conclude that the minimum expected total cost in fuzzy case is less than in the crisp case, which indicates that the fuzziness is very close to the actuality of life and gets minimum expected total cost less than the crisp case.

For the results of the numerical example, we note that when β increases, r^* increases, and thus Q^* decreases which indicate that the min E(TC) decreases.

Also, the different values of β lead to changes of Q^* in each distribution separately. But in all distributions we note that values of Q^* are almost fixed, due to the constraint on the varying order cost. Also, we note that when γ increases, min E(TC) decreases; this indicates that 70% of the shortages can be met at the lowest possible cost.

Finally, our study in particular provides the ample scope for further research and exploration. For instance, we have considered probabilistic mixture shortage inventory model under varying order cost constraint. This work can be further developed by considering an ample range of different assumptions and conditions represented in constraints and costs (constant or varying), such as varying two costs under two constraints or varying two costs under constraint or varying one cost under two constraints. Also, we can study some of the inventory models with the system multiechelon-multisource.

Table 3: The exact solutions and min E(TC) for model (I) and model (I_f) at Uniform distribution.

γ	β	Crisp case			Fuzzy case		
		Q^*	r^*	min E(TC)	Q^*	r^*	min E(TC)
0.1	0.1	297.124	17.0568	4067.24	250.44	18.0722	2623.71
	0.2	184.926	19.6971	2697.71	158.084	20.4323	1791.22
	0.3	123.756	21.4539	1955.07	107.098	21.978	1333.88
	0.4	87.7442	22.6221	1519.47	76.748	22.9994	1062.47
	0.5	65.1225	23.415	1246.58	57.4541	23.6935	890.457
	0.6	50.187	23.9661	1066.66	44.6672	24.1744	776.304
	0.7	39.8908	24.3597	942.705	35.7373	24.5207	696.802
	0.8	32.4803	24.6502	853.887	29.23	24.7787	639.579
0.3	0.1	297.121	15.5207	4048	250.42	16.8872	2612.58
	0.2	184.862	18.7021	2684.53	158.097	19.6747	1784.32
	0.3	123.725	20.7762	1946.25	107.066	21.4673	1328.91
	0.4	87.7476	22.1372	1513.44	76.7256	22.6354	1058.95
	0.5	65.132	23.0538	1242.15	57.475	23.421	888.053
	0.6	50.1549	23.6892	1062.94	44.6562	23.9647	774.318
	0.7	39.8284	24.1411	939.565	35.7041	24.3546	695.179
	0.8	32.4725	24.4703	851.602	29.2817	24.6397	638.328
0.7	0.1	297.1	10.0378	3979.21	250.426	12.9987	2576.66
	0.2	184.862	15.3252	2642.32	158.11	17.311	1762.56
	0.3	123.72	18.5524	1918.4	107.112	19.9169	1314.92
	0.4	87.7011	20.5852	1493.56	76.6763	21.5568	1048.64
	0.5	65.1351	21.9128	1227.92	57.4479	22.6277	880.563
	0.6	50.1559	22.8182	1052.06	44.6218	23.3576	768.557
	0.7	39.8823	23.4508	931.288	35.6999	23.873	690.713
	0.8	**32.4596**	**23.9138**	**844.584**	**29.3328**	**24.2447**	**634.709**

Notations

$\overline{D}$: A random variable denoting the demand rate per period
Q: A decision variable representing the order quantity per cycle
r: A decision variable representing the reorder point
L: The lead time between the placement of an order and its receipt
x: The continuous random variable representing the demand during L
$f(x)$: The probability density function of the lead time demand and (x) is its distribution function
$R(r)$: The probability of the shortage $= 1 - F(r) = \int_r^\infty f(x)\,dx$
$\overline{S}(r)$: The expected value of shortages per cycle $= \int_r^\infty (x - r) f(x)\,dx$
c_o: The order cost per unit
$C_o(Q) = c_o Q^{-\beta}$: The varying order cost per cycle
β: A constant real number selected to provide the best fit of estimated expected cost function
c_h: The holding cost per unit per period
c_s: The shortage cost per unit
c_b: The backorders cost per unit
c_l: The lost sales cost per unit
K: The limitation on the expected annual order cost
λ: The Lagrangian multiplier.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This research project was supported by a grant from the "Research Center of the Female Scientific and Medical Colleges," Deanship of Scientific Research, King Saud University.

References

[1] G. Hadley and T. M. Whitin, *Analysis of Inventory System*, Prentice Hall, Englewood Cliffs, NJ, USA, 1963.

[2] Z. T. Balkhi and L. Benkherouf, "A production lot size inventory model for deteriorating items and arbitrary production and demand rates," *European Journal of Operational Research*, vol. 92, no. 2, pp. 302–309, 1996.

[3] M. O. Abuo-El-Ata, H. A. Fergany, and M. F. El-Wakeel, "Probabilistic multi-item inventory model with varying order cost under two restrictions: a geometric programming approach," *International Journal of Production Economics*, vol. 83, no. 3, pp. 223–231, 2003.

[4] T. Vijayan and M. Kumaran, "Inventory models with a mixture of backorders and lost sales under fuzzy cost," *European Journal of Operational Research*, vol. 189, no. 1, pp. 105–119, 2008.

[5] H. A. Fergany, *Inventory models with demand-dependent units cost [Ph.D. dissertation]*, Faculty of Science, Tanta University, 1999.

[6] H. A. Fergany and M. E. El-Saadani, "Constrained probabilistic inventory model with continuous distributions and varying holding cost," *International Journal of Applied Mathematics*, vol. 17, pp. 53–67, 2005.

[7] A. F. Hala and F. E. Mona, "Constrained probabilistic lost sales inventory system with normal distribution and varying order cost," *Journal of Mathematics and Statistics*, vol. 2, no. 1, pp. 363–366, 2006.

[8] H. A. Fergany and M. F. El-Wakeel, "Constrained probabilistic lost sales inventory system with continuous distributions and varying order cost," *Journal of Association for the Advancement of Modelling and Simulation Techniques in Enterprises*, vol. 27, pp. 3–4, 2006.

[9] M. F. El-Wakeel, "Constrained backorders inventory system with varying order cost: lead time demand uniformly distributed," *Journal of King Saud University—Science*, vol. 24, no. 3, pp. 285–288, 2012.

[10] M. F. El-Wakeel and H. A. Fergany, "Constrained probabilistic continuous review inventory system with mixture shortage and stochastic lead time demand," *Advances in Natural Science*, vol. 6, no. 1, pp. 9–13, 2013.

[11] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[12] J. K. Syed and L. A. Aziz, "Fuzzy inventory model without shortages using signed distance method," *International Journal of Applied Mathematics and Information Sciences*, vol. 1, no. 2, pp. 203–209, 2007.

[13] N. Kazemi, E. Ehsani, and M. Y. Jaber, "An inventory model with backorders with fuzzy parameters and decision variables," *International Journal of Approximate Reasoning*, vol. 51, no. 8, pp. 964–972, 2010.

[14] O. A. Gawdt, *Some types of the probabilistic inventory models [Ph.D. thesis]*, Faculty of Science, Tanta University, 2011.

[15] H. A. Fergany and O. A. Gawdt, "Continuous review inventory model with mixture shortage under constraint involving crashing cost based on probabilistic triangular fuzzy numbers," *The Online Journal on Mathematics and Statistics*, vol. 2, no. 1, pp. 42–48, 2011.

[16] H. A. Fergany, N. A. El-Hefnawy, and O. M. Hollah, "Probabilistic periodic review <Q_M, N> inventory model using Lagrange technique and fuzzy adaptive particle swarm optimization," *Journal of Mathematics and Statistics*, vol. 10, no. 3, pp. 368–383, 2014.

[17] S. Kumar and U. S. Rajput, "Fuzzy inventory model for deteriorating items with time dependent demand and partial backlogging," *International Journal of Applied Mathematics*, vol. 6, no. 3, pp. 496–509, 2015.

[18] S. K. Indrajitsingha, P. N. Samanta, and U. K. Misra, "Fuzzy inventory model with shortages under fully backlogged using signed distance method," *International Journal for Research in Applied Science & Engineering Technology*, vol. 4, pp. 197–203, 2016.

[19] P. D. Patel, A. S. Gor, and P. Bhathawala, "Continuous review inventory model under fuzzy environment without backorder for deteriorating items," *International Journal of Applied Research*, vol. 2, no. 3, pp. 682–686, 2016.

[20] J.-S. Yao and K. Wu, "Ranking fuzzy numbers based on decomposition principle and signed distance," *Fuzzy Sets and Systems*, vol. 116, no. 2, pp. 275–288, 2000.

[21] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, McGraw-Hill, New York, NY, USA, 1995.

[22] S. Nahmias, *Production and Operations Analysis*, Irwin, Inc, Homewood, Ill, USA, 2nd edition, 1993.

[23] W. J. Fabrycky and J. Banks, *Procurement and Inventory Systems: Theory and Analysis*, Reinhold Publishing Corporation, New York, NY, USA, 1967.

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2016, Article ID 1479692, 13 pages
http://dx.doi.org/10.1155/2016/1479692

Research Article

Understanding Open Source Software Evolution Using Fuzzy Data Mining Algorithm for Time Series Data

Munish Saini,[1] Sandeep Mehmi,[2] and Kuljit Kaur Chahal[1]

[1]*Department of Computer Science, Guru Nanak Dev University, Amritsar, India*
[2]*Department of Computer Science, I.K.G. Punjab Technical University, Jalandhar, Punjab, India*

Correspondence should be addressed to Munish Saini; munish_1_saini@yahoo.co.in

Received 6 June 2016; Accepted 20 July 2016

Academic Editor: Gözde Ulutagay

Source code management systems (such as Concurrent Versions System (CVS), Subversion, and git) record changes to code repositories of open source software projects. This study explores a fuzzy data mining algorithm for time series data to generate the association rules for evaluating the existing trend and regularity in the evolution of open source software project. The idea to choose fuzzy data mining algorithm for time series data is due to the stochastic nature of the open source software development process. Commit activity of an open source project indicates the activeness of its development community. An active development community is a strong contributor to the success of an open source project. Therefore commit activity analysis along with the trend and regularity analysis for commit activity of open source software project acts as an important indicator to the project managers and analyst regarding the evolutionary prospects of the project in the future.

1. Introduction

Understanding software evolution in general and open source software (OSS) evolution in particular has been of wide interest in the recent past. A wide range of research studies have analysed OSS project evolution from different points of views such as growth [1], quality [2], and group dynamics [3]. However, there are a very few studies on commit activity in OSS projects. A commit is a change to a source code entity submitted by a developer through a source code management (SCM) system. SCM systems, such as Subversion (SVN) and git [4], manage the source code files of OSS systems and maintain log of each change (a.k.a. commit) made to the files. Committing is an important activity of the OSS development approach. Most of the OSS developers being volunteers, the success of these OSS projects is mainly determined by the committing activities of developers [5]. Commit activity indicates project activity which is further related to project success [6, 7]. Stakeholders of an OSS project, such as project managers, developers, and users, are interested in its future change behavior. Analysing the commit activity of an OSS project for finding trend and regularities in the evolution helps in indicating the future change behavior of the project and helps in decision making as far as project usage and management are concerned.

The OSS development is a stochastic process. Unlike the traditional development in which the environment is controlled, OSS development is based on contributions from volunteers who could not be forced to work even if something is of high priority for the project [1]. Along with this unplanned activity, there is a lack of planned documentation related to requirements and detailed design [8]. Classical time series techniques are inappropriate for analysis and forecasting of the data which involves random variables [8, 9]. Fuzzy time series can work for domains which involve uncertainty.

Commit activity of an OSS project is measured with the number of commits per month metric [5]. Kemerer and Slaughter [9] and Mockus and Votta [10] emphasize that the commits available in a SCM system (such as git [4]) can be used as a metric to study the evolution of OSS systems, and these studies motivate us too to choose number of commits per month as a metric to analyse and predict software evolution.

In this research work, the hybrid approach proposed by Chen et al. [11] (fuzzy theory along with data mining

algorithm) is used to generate linguistic rules from the time series data. In [11], Chen et al. specified the finding application for their algorithm and validating it as the future work. This present study considers both of these issues as one of the objectives. In this proposed work we divide the considered time series dataset into two data subsets, that is, training set and remaining set. We use the algorithm first to generate the association rules from the training set and then validate the accuracy and prediction capability of these rules on the remaining set. The high prediction accuracy indicates that the commits in the remaining set have regularity and trend in the number of commits performed for Eclipse CDT. The low prediction accuracy specifies that the commits on remaining set are not consistent with those of training set and there is irregularity and detrend in the commits performed.

Main objective of the present study is to explore the fuzzy data mining approach for time series used to generate the association rules for evaluating the existing trend and regularity in the evolution of OSS projects. As commit activity is a good indicator of continuous development activity of an OSS project, another objective of the present work is to develop a commit prediction model for OSS systems. Project managers, developers, and users can use the commit prediction model to understand the future commit activity of an OSS project and then plan schedule and allocate resources accordingly as per their role.

The rest of the paper is as follows. Section 2 presents the related work to study software evolution and the used fuzzy data mining technique. Section 3 explains the research methodology. Section 4 presents details of the experimental setup used for performing the study. Section 5 gives the results and analysis. Threats to validity of the results are explained in Section 6. Last section concludes the paper.

2. Related Work

The idea to analyse and predict the software evolution was seen initially in the late 1980s, when Yuen published his papers on the subject in a series of conferences on software maintenance [12–14]. He used time series analysis as a technique for software evolution prediction. Later, several studies used time series analysis for predicting software evolution. The software evolution metrics undertaken for prediction include the monthly number of changes [8, 9], change requests [15, 16], size and complexity [17, 18], defects [19, 20], clones [21], and maintenance effort [22].

Kemerer and Slaughter [9] looked at the evolution of two proprietary systems using two approaches: one based on the time series analysis (ARIMA) and the other based on a technique called sequence analysis. They found ARIMA models inappropriate for analysis as the dataset was largely random in nature. Antoniol et al. [21] presented an approach for monitoring and predicting evolution of software clones across subsequent versions of a software system (*mSQL*) using time series analysis (ARIMA).

Caprio et al. [17] used time series analysis to estimate size and complexity of the Linux Kernel. They used ARIMA model to predict evolution of the Linux Kernel by using dataset related to 68 stable releases of the software system.

Herraiz et al. [8] applied a stationery model based on time series analysis to the monthly number of changes in the CVS repository of Eclipse. Their model predicted the number of changes per month for the next three months. They employed Kernel smoothing to reduce noise, a lesson learned from Kemerer and Slaughter [9] study who could not get good results of ARIMA modelling for predicting number of changes, as they ignored noise present in the data.

Kenmei et al. [16] applied ARIMA to model and forecast change requests per unit of size of large open source projects. Data from three large open source projects, Mozilla, JBoss, and Eclipse, confirm the capability of the approach to effectively perform prediction and identify trends. They report the evidence that ARIMA models almost always outperform the predictive accuracy of simple models such as linear regression or random walk [16]. The benchmark models selected by Kenmei et al. [16] for evaluating the prediction accuracy of ARIMA model are not rigorous.

Raja et al. [20] did time series analysis of defect reports of eight open source software projects over a period of five years and found ARIMA (0, 1, 1) model to be useful for defect prediction. Kläs et al. [19] combined time series analysis with expert opinion to create prediction models for defects. They suggest that a hybrid model is more powerful than data models in the early phases of a project's life cycle.

Goulão et al. [15] used time series technique for long-term prediction of the overall number of change requests. They investigated the suitability of ARIMA model to predict the long-term fluctuation of all change requests for a project having seasonal patterns, such as Eclipse. They found that their ARIMA model is statistically more significant and outperforms the nonseasonal models.

Amin et al. [23] used the ARIMA model in place of software reliability growth model (SRGM) to predict the software reliability. SRGM has restrictive assumption on environment of the software under analysis. They specified that the ARIMA modelling is far better than SRGM approach as ARIMA is data oriented and cover all limitations of the previous approaches.

A perusal of the existing research in this area shows ARIMA modelling as the most frequently used prediction procedure. However, OSS development is a stochastic process. Unlike the traditional development in which the environment is controlled, OSS development is based on contributions from volunteers who could not be forced to work even if something is of high priority for the project [1]. Along with this unplanned activity, there is a lack of planned documentation related to requirements and detailed design [8].

Open source projects, without any tight organizational support, face many uncertainties. Uncertainty lies in an uncontrolled development environment such as availability of contributors at any point of time. Due to uncertainty, there is a large fluctuation in consecutive values (as observed in monthly commit data of Eclipse CDT in Figure 1). Most of the classical time series techniques are inappropriate for analysis and forecasting of the data which involve uncertainty [8, 9]. Research literature also indicates that ARIMA modelling can

Table 1: Descriptive statistics of the open source software projects.

Software	Origin date	Data collection date	Number of years	Number of authors	Total commits analysed
Eclipse CDT	6/27/2002	9/23/2013	11	135	26246

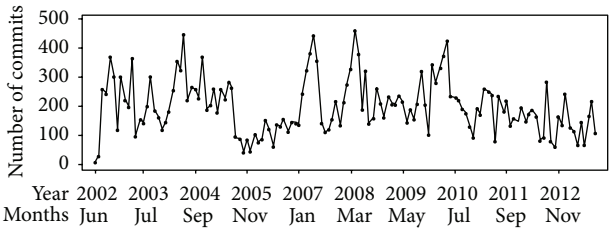

Figure 1: Eclipse CDT commit activity.

be useful when there is uncertainty in the data, but only after applying smoothing to reduce noise [24].

Hong et al. [25] introduced the fuzzy data mining algorithm for quantitative values. The algorithm extracts the useful knowledge from the transactional database having quantitative values. The algorithm combines the fuzzy set concept with that of Apriori algorithm.

Chen et al. [11] extended the work of Hong et al. and analysed the fuzzy data mining algorithm on time series data. They proposed an approach in which the concepts of fuzzy sets are used along with the data mining Apriori algorithm to generate linguistic association rules. They use the fuzzy membership function to convert the time series data to fuzzy set and then apply the Apriori algorithm to generate association rules accordingly. They specified the validation and finding of applications for their algorithm as the future work. So, with the continuation of their work, we are using algorithm for analysing the commit activity of open source software project for finding existing regularity and trend in the commit activity of OSS project.

Suresh and Raimond [26] extended the work of Chen et al. [11]. They proposed a new algorithm called extended fuzzy frequent pattern algorithm for analysing the time series data. The association rules are generated without generating the candidate sets.

This paper uses a fuzzy data mining approach on time series [11] to analyse the commit activity in open source software projects. The research questions that this study aims to answer are (1) analysing the commit activity of OSS project for finding the trend and regularity and (2) validating the fuzzy data mining algorithm on OSS project data.

3. Methodology

The objective of this empirical study is to investigate the suitability of the fuzzy data mining method to analyse the number of commits per month as a software system evolves for finding the regularity and trend. This section describes the data collection process, basic concepts of fuzzy time series, and the fuzzy data mining method.

3.1. Data Collection. The development repository of open source software project (Eclipse CDT) is obtained from GIT Hub [27]. A repository is downloaded by making the clone of the original repository onto the local machine by using GIT Bash [4]. A script is written in JAVA to fetch the number of commits per month for the observation period for all the software projects. The descriptive statistics about the development repositories of all software projects is shown in Table 1.

Eclipse [28] is an integrated development environment. Eclipse has base workspace and extendable plug-in for customizing the development environment. It is used to develop application in different languages such as JAVA, C/C++, COBOL, and PHP, by using available plug-in. The two variations Eclipse SDK and Eclipse CDT are well known for developing applications. Eclipse SDK is compatible with JAVA and used by JAVA developers for building project on Eclipse platform. Eclipse CDT provides C/C++ development tooling [29]. Eclipse CDT allows developing application in C/C++ using Eclipse. Eclipse CDT provides various features [30] such as full featured editor, debugging, refactoring, parser, and indexes. In this study we consider Eclipse CDT only. The number of commits for each month from 6/27/2002 to 9/23/2013 is arranged month-wise to form a time series (for 136 months) shown in Figure 1.

3.2. Basics of Fuzzy Set Theory and Fuzzy Time Series. The concept of fuzzy set was introduced by Zadeh [31] in 1965. It was the extension of classical set theory. The fuzzy set is characterized by degree of membership function [31]. The membership function can be of various forms such as triangular, L-function, R-function, and trapezoidal function used depending upon the application and requirement. The present study uses both L and R membership function to define the values of fuzzy variable.

3.3. Apriori Algorithm. Apriori algorithm [32, 33] is one of the classical algorithms proposed by R. Srikant and R. Agrawal in 1994 for finding frequent patterns for generating association rules. Apriori employs an iterative approach known as level-wise search, where k-itemsets are used to explore $(k + 1)$-itemsets.

Apriori algorithm is executed in two steps. Firstly it retrieves all the frequent itemsets whose support is not smaller than the minimum support (min_sup). The first step further consists of join and pruning action. In joining, the candidate set C_k is produced by joining L_{k-1} with itself. In pruning, the candidate sets are pruned by applying the Apriori property; that is, all the nonempty subsets of frequent itemset must also be frequent.

The pseudocode for generation of frequent itemsets is as follows:

C_k: Candidate itemset of size k

L_k: Frequent itemset of size k

L_1 = frequent 1-itemset

For ($k = 1$; $L_k! = \Phi$; $k++$)
{
C_{k+1} = Join L_k with L_k to generate C_{k+1};
L_{k+1} = Candidate in C_{k+1} with support greater than or equal to min support;
}
End;
Return L_k;

Next, it uses the frequent itemsets to generate the strong association rules satisfying the minimum confidence (min_conf) threshold. The pseudocode for generation of strong association rules is as follows:

Input:
Frequent Itemset, L
Minimum confidence threshold, min_conf
Output: Strong association rules, R
$R = \Phi$
for each frequent itemset I in L
{
for each non-empty subset s of I
{
conf ($S \rightarrow I - S$) = support_count (I)/support_count (S)
if conf >= min_conf
{
// generate strong association rule
Rule $r =$ "$s \rightarrow (I - S)$"
$R = RU\{r\}$
}
}
}

3.4. Fuzzy Data Mining Algorithm for Time Series Data. Chen et al. [11] extended the work of Hong et al. [25] and proposed the fuzzy data mining for time series data. The time series data of K points are entered as input along with the predefined minimum support λ, minimum confidence α, and window size of w.

The input data is first converted to generate ($K - w + 1$) sequences; each subsequence has w elements. The fuzzy membership function is used to convert each data item into the equivalent fuzzy set. The Apriori algorithm is used to mine frequent fuzzy sets. Moreover, the data reduction method is used to remove the redundant data items.

The association rules are generated in the same way as generated in Apriori algorithm. The stepwise process of the fuzzy Apriori algorithm is given below.

Input:
Time series with K data points
Membership function values h
Minimum support λ
Minimum confidence α
Sliding window size w
Fuzzy set f_p
Output: Set of fuzzy association rules

Step 1. Convert the time series data into ($K-w+1$) sequences, where each sequence has maximum of w elements. Suppose we assume $w = 5$; then each sequence has maximum of 5 elements. The elements of subsequence are referred to as data variable and given as A_1, A_2, A_3, A_4, and A_5, respectively.

Step 2. Apply fuzzy membership function on elements of time series to generate fuzzy set (f_p).

Step 3. Based on the membership function and its user defined level (suppose low, middle, and high), each data variable after conversion to fuzzy item lies in different user defined levels (such as $A_{1.\text{Low}}$, $A_{2.\text{Middle}}$, and $A_{3.\text{High}}$ referred to as fuzzy items).

Step 3. Calculate the scalar cardinality count of each fuzzy item of subsequences.

Step 4. Compare the total scalar cardinality count of each subsequence with the minimum support value. The sequences with value greater than or equal to α are kept in L_1.

The support value of subsequence is generated as

$$\text{Support value} = \frac{\text{Count}}{K - w + 1}. \quad (1)$$

Step 5. If L_1 = NULL, then exit; else do the following step for $r = 1$ to K.

Step 6. Join L_r with L_r to generate candidate ($r + 1$) fuzzy itemset (i.e., C_{r+1}) (similar to Apriori algorithm except not joining the items generated from same order of data point and join is possible only if ($r - 1$) data items in both sets are the same).

(i) Calculate the fuzzy value of each candidate fuzzy itemset by using fuzzy set theory, that is, Min ($f_1 \Lambda f_2 \Lambda \ldots f_k$).

(ii) Count the scalar cardinality of each fuzzy candidate itemset.

(iii) If count is more than or equal to α, then put it in L_{r+1} and calculate its support value as

$$\text{Support value} = \frac{\text{Count}}{K - w + 1}. \quad (2)$$

Step 7. If L_{r+1} = NULL, then exit; otherwise go to Step 6 again.

Step 8. Remove redundant large itemset (i.e., by shifting each large itemsets ($I_1, I_2, I_3, \ldots, I_q$) into ($I_1', I_2', I_3', \ldots, I_q'$) such that fuzzy region R_{11} becomes R_{11}' when shifted).

Table 2: Fuzzy candidate item sets in C_2.

$A_{1.\text{Low}} \cap A_{2.\text{Low}}$	$A_{1.\text{Low}} \cap A_{2.\text{Middle}}$	$A_{1.\text{Low}} \cap A_{3.\text{Low}}$	$A_{1.\text{Low}} \Lambda A_{3.\text{Middle}}$	$A_{1.\text{Low}} \Lambda A_{4.\text{Low}}$
$A_{1.\text{Low}} \cap A_{4.\text{Middle}}$	$A_{1.\text{Low}} \cap A_{5.\text{Low}}$	$A_{1.\text{Low}} \cap A_{5.\text{Middle}}$	$A_{1.\text{Middle}} \cap A_{2.\text{Low}}$	$A_{1.\text{Middle}} \cap A_{2.\text{Middle}}$
$A_{1.\text{Middle}} \cap A_{3.\text{Low}}$	$A_{1.\text{Middle}} \cap A_{3.\text{Middle}}$	$A_{1.\text{Middle}} \cap A_{4.\text{Low}}$	$A_{1.\text{Middle}} \cap A_{4.\text{Middle}}$	$A_{1.\text{Middle}} \cap A_{5.\text{Low}}$
$A_{1.\text{Middle}} \cap A_{5.\text{Middle}}$	$A_{2.\text{Low}} \cap A_{3.\text{Low}}$	$A_{2.\text{Low}} \cap A_{3.\text{Middle}}$	$A_{2.\text{Low}} \cap A_{4.\text{Low}}$	$A_{2.\text{Low}} \cap A_{4.\text{Middle}}$
$A_{2.\text{Low}} \cap A_{5.\text{Low}}$	$A_{2.\text{Low}} \cap A_{5.\text{Middle}}$	$A_{2.\text{Middle}} \cap A_{3.\text{Low}}$	$A_{2.\text{Middle}} \cap A_{3.\text{Middle}}$	$A_{2.\text{Middle}} \cap A_{4.\text{Low}}$
$A_{2.\text{Middle}} \cap A_{4.\text{Middle}}$	$A_{2.\text{Middle}} \cap A_{5.\text{Low}}$	$A_{2.\text{Middle}} \cap A_{5.\text{Middle}}$	$A_{3.\text{Low}} \cap A_{4.\text{Low}}$	$A_{3.\text{Low}} \cap A_{4.\text{Middle}}$
$A_{3.\text{Low}} \cap A_{5.\text{Low}}$	$A_{3.\text{Low}} \cap A_{5.\text{Middle}}$	$A_{3.\text{Middle}} \cap A_{4.\text{Low}}$	$A_{3.\text{Middle}} \cap A_{4.\text{Middle}}$	$A_{3.\text{Middle}} \cap A_{5.\text{Low}}$
$A_{3.\text{Middle}} \cap A_{5.\text{Middle}}$	$A_{4.\text{Low}} \cap A_{5.\text{Low}}$	$A_{4.\text{Low}} \cap A_{5.\text{Middle}}$	$A_{4.\text{Middle}} \cap A_{5.\text{Low}}$	$A_{4.\text{Middle}} \cap A_{5.\text{Middle}}$

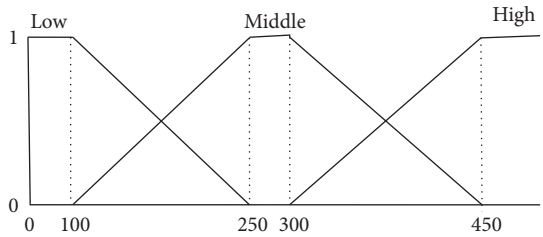

Figure 2: Membership function.

Step 9. Generate the association rules using Apriori rule generation method and calculate the confidence of each rule. The only variation is that in place of normal data itemset we have fuzzy itemset, so concept of intersection is used in rule not union. If confidence value is not less than minimum confidence α, then keep the rule; otherwise reject.

4. Experimental Setup

The experiment is performed on x86 Family 6 Model 15 Stepping 6 GenuineIntel ~2131 Mhz Processor 1 GB RAM, Microsoft Windows XP Professional operating system, version 5.1.2600 Service Pack 3 Build 2600, 240 GB hard disk. The fuzzy data mining algorithm for time series data is implemented using JAVA.

5. Results and Analysis

The dataset is divided into two subsets: training dataset (120 months) and remaining dataset (16 months). The complete process of generating the association rule using fuzzy data mining algorithm is performed in two steps.

(A) The association rules are generated on training dataset using fuzzy data mining for time series data algorithm.

(B) The validation of generated association rule is done using training and remaining dataset.

5.1. Generation of Association Rules from Training Dataset. The fuzzy data mining algorithm of time series [11] is applied on training dataset to generate association rule. We assumed $w = 5$, $\lambda = 25\%$ (25/100 $*$ number of subsequences), and $\alpha = 65\%$; membership function and its values are shown in Figure 2. These assumptions are made by referring to and understanding the concepts of fuzzy data mining algorithm given in [11]. The commits are divided according to the level of activity. The commits in the range of 0–100, 250–300, and 450 onwards indicate the *low*, *middle* (average), and *high* level of activity, respectively. The level of commit activity indicates the amount of work done in a commit.

(i) Generation of Subsequences

We have $K = 120$ (as number of months)

$w = 5$ (window size)

The number of subsequences is generated as $(K-w+1)$

Therefore number of subsequences $= (K - w + 1) = (120 - 5 + 1) = 116$

All generated subsequences for $K = 120$ (shown in Table 8 of the Appendix).

(ii) Transformation of Data to Fuzzy Sets. In this step, we transform the commit activity data into fuzzy set (using membership function) and count the scalar cardinality of each data variable (shown in Table 9 of the Appendix).

All these data variables are considered as candidate itemsets (C_1).

For generation of L_1, data variables having count more than $\lambda = 25\%$ (25/100 $*$ 116) $= 29$ are considered.

It is found that L_1 contain $A_{1.\text{Low}}$, $A_{1.\text{Middle}}$, $A_{2.\text{Low}}$, $A_{2.\text{Middle}}$, $A_{3.\text{Low}}$, $A_{3.\text{Middle}}$, $A_{4.\text{Low}}$, $A_{4.\text{Middle}}$, $A_{5.\text{Low}}$, and $A_{5.\text{Middle}}$.

Further, calculate support value of each candidate itemset using

$$\text{Support value} = \frac{\text{Count}}{K - w + 1}. \tag{3}$$

(iii) Generation of C_2 and L_2. For generating C_2 (shown in Table 2), join L_1 with L_1, not joining the items generated from same order of data points; that is, $A_{1.\text{Low}}$ join with $A_{1.\text{Middle}}$ is not allowed, similar to all others also. Meanwhile, joining all the properties of Apriori algorithm is used.

Next, use Min $(f_1 \cap f_2)$ function to find the value of each of the candidate fuzzy sets. Count the scalar cardinality of each of the candidate sets in C_2.

Table 3: Fuzzy item sets in L_2.

$A_{1.\text{Low}} \cap A_{2.\text{Low}}$	$A_{1.\text{Low}} \cap A_{3.\text{Low}}$	$A_{1.\text{Low}} \cap A_{4.\text{Middle}}$	$A_{1.\text{Middle}} \cap A_{2.\text{Middle}}$	$A_{1.\text{Middle}} \cap A_{3.\text{Middle}}$
$A_{1.\text{Middle}} \cap A_{4.\text{Middle}}$	$A_{1.\text{Middle}} \cap A_{5.\text{Low}}$	$A_{1.\text{Middle}} \cap A_{5.\text{Middle}}$	$A_{2.\text{Low}} \cap A_{3.\text{Low}}$	$A_{2.\text{Low}} \cap A_{4.\text{Low}}$
$A_{2.\text{Middle}} \cap A_{3.\text{Middle}}$	$A_{2.\text{Middle}} \cap A_{4.\text{Middle}}$	$A_{2.\text{Middle}} \cap A_{5.\text{Middle}}$	$A_{3.\text{Low}} \cap A_{4.\text{Low}}$	$A_{3.\text{Low}} \cap A_{5.\text{Low}}$
$A_{3.\text{Middle}} \Lambda A_{4.\text{Middle}}$	$A_{3.\text{Middle}} \cap A_{5.\text{Middle}}$	$A_{4.\text{Low}} \cap A_{5.\text{Low}}$	$A_{4.\text{Middle}} \cap A_{5.\text{Middle}}$	

Table 4: Fuzzy item sets in L_6.

$A_{1.\text{Middle}} \cap A_{2.\text{Middle}} \cap A_{3.\text{Middle}}$	$A_{1.\text{Middle}} \cap A_{2.\text{Middle}} \cap A_{4.\text{Middle}}$	$A_{1.\text{Middle}} \cap A_{2.\text{Middle}} \cap A_{5.\text{Middle}}$
$A_{1.\text{Middle}} \cap A_{3.\text{Middle}} \cap A_{4.\text{Middle}}$	$A_{1.\text{Middle}} \cap A_{3.\text{Middle}} \cap A_{5.\text{Middle}}$	$A_{1.\text{Middle}} \cap A_{4.\text{Middle}} \cap A_{5.\text{Middle}}$
$A_{2.\text{Middle}} \cap A_{3.\text{Middle}} \cap A_{4.\text{Middle}}$	$A_{2.\text{Middle}} \cap A_{3.\text{Middle}} \cap A_{5.\text{Middle}}$	$A_{2.\text{Middle}} \cap A_{4.\text{Middle}} \cap A_{5.\text{Middle}}$
$A_{3.\text{Middle}} \cap A_{4.\text{Middle}} \cap A_{5.\text{Middle}}$		

Table 5: Largest fuzzy item sets generated.

$A_{1.\text{Middle}} \cap A_{2.\text{Middle}} \cap A_{3.\text{Middle}}$	$A_{1.\text{Middle}} \cap A_{2.\text{Middle}} \cap A_{4.\text{Middle}}$	$A_{1.\text{Middle}} \cap A_{2.\text{Middle}} \cap A_{5.\text{Middle}}$
$A_{1.\text{Middle}} \cap A_{3.\text{Middle}} \cap A_{4.\text{Middle}}$	$A_{1.\text{Middle}} \cap A_{3.\text{Middle}} \cap A_{5.\text{Middle}}$	$A_{1.\text{Middle}} \cap A_{4.\text{Middle}} \cap A_{5.\text{Middle}}$

The candidate fuzzy sets with a value not less than the threshold value are kept in L_2 and also find their support value. Now, L_2 contain fuzzy itemsets shown in Table 3.

(iv) Generation of C_3 and L_3. For generating C_3, join L_2 with L_2, not joining the items generated from same order of data points. In case of C_3, join is possible between only those fuzzy sets where at least one data item is common in both. After generation of C_3, only those fuzzy itemsets are put in L_3 (shown in Table 4) whose count is not less than threshold value.

(v) Generation of C_4. Join L_3 with L_3, not joining the items generated from same order of data point. Join is possible only for those fuzzy sets where at least two data items are common in both.

After generation of C_4, it is found that no element has count more than threshold value; therefore L_4 = Null.

(vi) Removal of Redundant Large Itemsets. Remove the redundant large itemsets from L_3 using Step 8 of the algorithm described in Section 3.4.

After applying Step 8 of algorithm, we are left with these large itemsets (shown in Table 5).

(vii) Generate Association Rules. Generate the association rules from these large itemsets using Step 8 of the algorithm described in Section 3.4. All the generated rules are shown in Table 6 where strong rules having count more than threshold confidence are marked as bold.

In this experiment we use $\alpha = 65\%$; it means only those rules are valid (and are called strong association rules) and have support value not less than 65%. We found 18 rules in this case; each rule acts as the knowledge base for the project manager and developers of the project. For example, the generated rule $\mathbf{A_{1.Middle}} \cap \mathbf{A_{2.Middle}} \rightarrow \mathbf{A_{3.Middle}}$ specifies that if the value of first and second data point lies in the middle, then there is high probability that the third data point has middle value also. All these rules act as a precise and compact knowledge for project manager and analyst.

5.2. Validation of Association Rules Using Training and Remaining Dataset

(i) Validation on Training Dataset. It is found that 65% of the transactions in the Eclipse CDT training set follow these rules. These rules act as a compact and concrete knowledge of this data.

(ii) Validation on Remaining Dataset. All generated association rules are tested on remaining dataset values to find the regularity and trend in the number of commits analysed to find whether the commits are performed at same rate or not. If the rate is same, then it means Eclipse has consistent growth; otherwise there exists a variation in the considered software growth. It is found that in case of remaining set the applicability of these rules decreases. This thing is again verified by generating the association rules from the remaining set. The generated rules from remaining dataset specify that the data in the remaining dataset is more towards lower range of commits performed.

Following are the factors due to which this variation in the commit rate is found:

(a) Most of the values in the remaining dataset are towards *low* range. This point is verified by finding the largest frequent fuzzy itemsets and then generating the association rules from the remaining dataset by using fuzzy data mining for time series data algorithm. The largest frequent itemset found is shown in Table 7.

The generated frequent itemsets consist of only data points with low range. Hence most of the entries of commits in remaining datasets are probably *low*. This specifies that the numbers of commits analysed from 6/1/2012 to 9/23/2013 is less as compared to the number of commits analysed in the training dataset. It specifies the less activity in the development of considered software growth in this period.

Table 6: Calculated confidence value of various association rules.

$\mathbf{A_{1.Middle} \cap A_{2.Middle} \rightarrow A_{3.Middle}}$	**0.741**	**74%**
$A_{3.Middle} \rightarrow A_{1.Middle} \cap A_{2.Middle}$	0.522	52%
$\mathbf{A_{1.Middle} \cap A_{3.Middle} \rightarrow A_{2.Middle}}$	**0.781**	**78%**
$A_{2.Middle} \rightarrow A_{1.Middle} \cap A_{3.Middle}$	0.526	53%
$\mathbf{A_{2.Middle} \cap A_{3.Middle} \rightarrow A_{1.Middle}}$	**0.735**	**74%**
$A_{1.Middle} \rightarrow A_{2.Middle} \cap A_{3.Middle}$	0.529	53%
$\mathbf{A_{1.Middle} \cap A_{2.Middle} \rightarrow A_{4.Middle}}$	**0.692**	**69%**
$A_{4.Middle} \rightarrow A_{1.Middle} \cap A_{2.Middle}$	0.491	49%
$\mathbf{A_{1.Middle} \cap A_{4.Middle} \rightarrow A_{2.Middle}}$	**0.762**	**76%**
$A_{2.Middle} \rightarrow A_{1.Middle} \cap A_{4.Middle}$	0.491	49%
$\mathbf{A_{2.Middle} \cap A_{4.Middle} \rightarrow A_{1.Middle}}$	**0.724**	**72%**
$A_{1.Middle} \rightarrow A_{2.Middle} \cap A_{4.Middle}$	0.494	49%
$\mathbf{A_{1.Middle} \cap A_{2.Middle} \rightarrow A_{5.Middle}}$	**0.695**	**69%**
$A_{5.Middle} \rightarrow A_{1.Middle} \cap A_{2.Middle}$	0.497	50%
$\mathbf{A_{1.Middle} \cap A_{5.Middle} \rightarrow A_{2.Middle}}$	**0.737**	**74%**
$A_{2.Middle} \rightarrow A_{1.Middle} \cap A_{5.Middle}$	0.493	49%
$\mathbf{A_{2.Middle} \cap A_{5.Middle} \rightarrow A_{1.Middle}}$	**0.759**	**76%**
$A_{1.Middle} \rightarrow A_{2.Middle} \cap A_{5.Middle}$	0.496	50%
$\mathbf{A_{1.Middle} \cap A_{3.Middle} \rightarrow A_{4.Middle}}$	**0.742**	**74%**
$A_{4.Middle} \rightarrow A_{1.Middle} \cap A_{3.Middle}$	0.499	50%
$\mathbf{A_{1.Middle} \cap A_{4.Middle} \rightarrow A_{3.Middle}}$	**0.775**	**78%**
$A_{3.Middle} \rightarrow A_{1.Middle} \cap A_{4.Middle}$	0.496	50%
$\mathbf{A_{3.Middle} \cap A_{4.Middle} \rightarrow A_{1.Middle}}$	**0.691**	**69%**
$A_{1.Middle} \rightarrow A_{3.Middle} \cap A_{4.Middle}$	0.502	50%
$\mathbf{A_{1.Middle} \cap A_{3.Middle} \rightarrow A_{5.Middle}}$	**0.751**	**75%**
$A_{5.Middle} \rightarrow A_{1.Middle} \cap A_{3.Middle}$	0.510	51%
$\mathbf{A_{1.Middle} \cap A_{5.Middle} \rightarrow A_{3.Middle}}$	**0.757**	**76%**
$A_{3.Middle} \rightarrow A_{1.Middle} \cap A_{5.Middle}$	0.502	50%
$\mathbf{A_{3.Middle} \cap A_{5.Middle} \rightarrow A_{1.Middle}}$	**0.738**	**74%**
$A_{1.Middle} \rightarrow A_{3.Middle} \cap A_{5.Middle}$	0.509	51%
$\mathbf{A_{1.Middle} \cap A_{4.Middle} \rightarrow A_{5.Middle}}$	**0.792**	**79%**
$A_{5.Middle} \rightarrow A_{1.Middle} \cap A_{4.Middle}$	0.515	51%
$\mathbf{A_{1.Middle} \cap A_{5.Middle} \rightarrow A_{4.Middle}}$	**0.764**	**76%**
$A_{4.Middle} \rightarrow A_{1.Middle} \cap A_{5.Middle}$	0.511	51%
$\mathbf{A_{4.Middle} \cap A_{5.Middle} \rightarrow A_{1.Middle}}$	**0.715**	**71%**
$A_{1.Middle} \rightarrow A_{4.Middle} \cap A_{5.Middle}$	0.513	51%

Table 7: Largest frequent fuzzy item set for remaining dataset.

$A_{1.Low} \cap A_{2.Low} \cap A_{3.Low} \cap A_{4.Low}$
$A_{1.Low} \cap A_{2.Low} \cap A_{3.Low} \cap A_{5.Low}$
$A_{1.Low} \cap A_{2.Low} \cap A_{4.Low} \cap A_{5.Low}$
$A_{1.Low} \cap A_{3.Low} \cap A_{4.Low} \cap A_{5.Low}$

(b) Factors like the number of active users and number of files changed (addition, deletion, and modification) are less.

6. Discussion

The fuzzy data mining algorithm for time series data [11] allows efficient mining of the association rules from the large

Table 8: Generated subsequences.

Sp	Subsequences				
1	9	25	252	240	361
2	25	252	240	361	298
3	252	240	361	298	118
4	240	361	298	118	294
5	361	298	118	294	219
6	298	118	294	219	193
7	118	294	219	193	359
8	294	219	193	359	96
9	219	193	359	96	151
10	193	359	96	151	141
11	359	96	151	141	198
12	96	151	141	198	296
13	151	141	198	296	183
14	141	198	296	183	160
15	198	296	183	160	117
16	296	183	160	117	139
17	183	160	117	139	179
18	160	117	139	179	255
19	117	139	179	255	351
20	139	179	255	351	320
21	179	255	351	320	440
22	255	351	320	440	219
23	351	320	440	219	260
24	320	440	219	260	254
25	440	219	260	254	227
26	219	260	254	227	363
27	260	254	227	363	185
28	254	227	363	185	200
29	227	363	185	200	254
30	363	185	200	254	175
31	185	200	254	175	253
32	200	254	175	253	221
33	254	175	253	221	281
34	175	253	221	281	263
35	253	221	281	263	87
36	221	281	263	87	83
37	281	263	87	83	42
38	263	87	83	42	80
39	87	83	42	80	44
40	83	42	80	44	99
41	42	80	44	99	71
42	80	44	99	71	83
43	44	99	71	83	150
44	99	71	83	150	120
45	71	83	150	120	60
46	83	150	120	60	132
47	150	120	60	132	127
48	120	60	132	127	151
49	60	132	127	151	109
50	132	127	151	109	141

Table 8: Continued.

Sp	Subsequences				
51	127	151	109	141	141
52	151	109	141	141	135
53	109	141	141	135	242
54	141	141	135	242	320
55	141	135	242	320	380
56	135	242	320	380	438
57	242	320	380	438	355
58	320	380	438	355	141
59	380	438	355	141	107
60	438	355	141	107	114
61	355	141	107	114	153
62	141	107	114	153	212
63	107	114	153	212	133
64	114	153	212	133	212
65	153	212	133	212	273
66	212	133	212	273	326
67	133	212	273	326	455
68	212	273	326	455	378
69	273	326	455	378	186
70	326	455	378	186	317
71	455	378	186	317	139
72	378	186	317	139	155
73	186	317	139	155	257
74	317	139	155	257	208
75	139	155	257	208	162
76	155	257	208	162	229
77	257	208	162	229	205
78	208	162	229	205	203
79	162	229	205	203	234
80	229	205	203	234	214
81	205	203	234	214	144
82	203	234	214	144	185
83	234	214	144	185	153
84	214	144	185	153	206
85	144	185	153	206	315
86	185	153	206	315	204
87	153	206	315	204	103
88	206	315	204	103	336
89	315	204	103	336	278
90	204	103	336	278	329
91	103	336	278	329	370
92	336	278	329	370	419
93	278	329	370	419	230
94	329	370	419	230	224
95	370	419	230	224	217
96	419	230	224	217	188
97	230	224	217	188	176
98	224	217	188	176	129
99	217	188	176	129	91
100	188	176	129	91	188

Table 8: Continued.

Sp	Subsequences				
101	176	129	91	188	169
102	129	91	188	169	256
103	91	188	169	256	245
104	188	169	256	245	237
105	169	256	245	237	74
106	256	245	237	74	228
107	245	237	74	228	179
108	237	74	228	179	212
109	74	228	179	212	128
110	228	179	212	128	154
111	179	212	128	154	150
112	212	128	154	150	192
113	128	154	150	192	148
114	154	150	192	148	171
115	150	192	148	171	181
116	192	148	171	181	163

dataset. These generated rules help in finding the regularity and existing trend for OSS projects. We have used the commit activity data of Eclipse CDT. The commits in repository are directly related to the activities such as file or code changed, deleted, or modified. By analysing the trend in commits data, we can interpret the development or evolution activity of the considered software. In the above experiment, the original dataset is divided into two subdatasets (training and remaining dataset).

The generated association rules from training set allow analysing the regularity and trends in the commits of OSS project. These rules also help in predicting and analysing the future evolution or development activity of the Eclipse CDT. The generated rules are validated on the remaining dataset to find its applicability. It is found that applicability of these rules on remaining set decreases. The results of the algorithm indicate that the commit activity of remaining dataset has *low* activity range. There may be other factors also which are the cause of this decrease behavior in the number of commits. These may include factors like the number of active users and number of files changed (addition, deletion, and modification) which are less.

7. Threats to Validity

This section discusses the threats to validity of the study.

Construct validity threats concern the relationship between theory and observation. These threats can be mainly due to the fact that we assumed all the commits posted in the revision control tool *git* [4]. Any changes performed in the source code, but not logged through the tool, may not have become part of the study.

Internal validity concerns the selection of subject systems and the analysis methods. This study uses a *month* as the unit of measure for tracking the types of change activities. In the future, we would like to use more natural and insightful partition based on major/minor versions of the OSS project

Table 9: Transformation of data elements into fuzzy set.

Count	47.595	60.227	8.178	47.275	60.547	8.178	46.802	61.02	8.178	47.262	60.56	8.178	47.775	60.047	8.178
Sp	$A_{1.Low}$	$A_{1.Middle}$	$A_{1.High}$	$A_{2.Low}$	$A_{2.Middle}$	$A_{2.High}$	$A_{3.Low}$	$A_{3.Middle}$	$A_{3.High}$	$A_{4.Low}$	$A_{4.Middle}$	$A_{4.High}$	$A_{5.Low}$	$A_{5.Middle}$	$A_{5.High}$
1	1	0	0	1	0	0	0	1	0	0.067	0.933	0	0	0.593	0.407
2	1	0	0	0	1	0	0.067	0.933	0	0	0.593	0.407	0	1	0
3	0	1	0	0.067	0.933	0	0	0.593	0.407	0	1	0	0.88	0.12	0
4	0.067	0.933	0	0	0.593	0.407	0	1	0	0.88	0.12	0	0	1	0
5	0	0.593	0.407	0	1	0	0.88	0.12	0	0	1	0	0.207	0.793	0
6	0	1	0	0.88	0.12	0	0	1	0	0.207	0.793	0	0.38	0.62	0
7	0.88	0.12	0	0	1	0	0.207	0.793	0	0.38	0.62	0	0	0.607	0.393
8	0	1	0	0.207	0.793	0	0.38	0.62	0	0	0.607	0.393	1	0	0
9	0.207	0.793	0	0.38	0.62	0	0	0.607	0.393	1	0	0	0.66	0.34	0
10	0.38	0.62	0	0	0.607	0.393	1	0	0	0.66	0.34	0	0.727	0.273	0
11	0	0.607	0.393	1	0	0	0.66	0.34	0	0.727	0.273	0	0.347	0.653	0
12	1	0	0	0.66	0.34	0	0.727	0.273	0	0.347	0.653	0	0	1	0
13	0.66	0.34	0	0.727	0.273	0	0.347	0.653	0	0	1	0	0.447	0.553	0
14	0.727	0.273	0	0.347	0.653	0	0	1	0	0.447	0.553	0	0.6	0.4	0
15	0.347	0.653	0	0	1	0	0.447	0.553	0	0.6	0.4	0	0.887	0.113	0
16	0	1	0	0.447	0.553	0	0.6	0.4	0	0.887	0.113	0	0.74	0.26	0
17	0.447	0.553	0	0.6	0.4	0	0.887	0.113	0	0.74	0.26	0	0.473	0.527	0
18	0.6	0.4	0	0.887	0.113	0	0.74	0.26	0	0.473	0.527	0	0	1	0
19	0.887	0.113	0	0.74	0.26	0	0.473	0.527	0	0	1	0	0	0.66	0.34
20	0.74	0.26	0	0.473	0.527	0	0	1	0	0	0.66	0.34	0	0.867	0.133
21	0.473	0.527	0	0	1	0	0	0.66	0.34	0	0.867	0.133	0	0.067	0.933
22	0	1	0	0	0.66	0.34	0	0.867	0.133	0	0.067	0.933	0.207	0.793	0
23	0	0.66	0.34	0	0.867	0.133	0	0.067	0.933	0.207	0.793	0	0	1	0
24	0	0.867	0.133	0	0.067	0.933	0.207	0.793	0	0	1	0	0	1	0
25	0	0.067	0.933	0.207	0.793	0	0	1	0	0	1	0	0.153	0.847	0
26	0.207	0.793	0	0	1	0	0	1	0	0.153	0.847	0	0	0.58	0.42
27	0	1	0	0	1	0	0.153	0.847	0	0	0.58	0.42	0.433	0.567	0
28	0	1	0	0.153	0.847	0	0	0.58	0.42	0.433	0.567	0	0.333	0.667	0
29	0.153	0.847	0	0	0.58	0.42	0.433	0.567	0	0.333	0.667	0	0	1	0
30	0	0.58	0.42	0.433	0.567	0	0.333	0.667	0	0	1	0	0.5	0.5	0
31	0.433	0.567	0	0.333	0.667	0	0	1	0	0.5	0.5	0	0	1	0
32	0.333	0.667	0	0	1	0	0.5	0.5	0	0	1	0	0.193	0.807	0
33	0	1	0	0.5	0.5	0	0	1	0	0.193	0.807	0	0	1	0
34	0.5	0.5	0	0	1	0	0.193	0.807	0	0	1	0	0	1	0
35	0	1	0	0.193	0.807	0	0	1	0	0	1	0	1	0	0
36	0.193	0.807	0	0	1	0	0	1	0	1	0	0	1	0	0
37	0	1	0	0	1	0	1	0	0	1	0	0	1	0	0
38	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0
39	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
40	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
41	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0

Table 9: Continued.

Count	47.595	60.227	8.178	47.275	60.547	8.178	46.802	61.02	8.178	47.262	60.56	8.178	47.775	60.047	8.178
Sp	$A_{1.Low}$	$A_{1.Middle}$	$A_{1.High}$	$A_{2.Low}$	$A_{2.Middle}$	$A_{2.High}$	$A_{3.Low}$	$A_{3.Middle}$	$A_{3.High}$	$A_{4.Low}$	$A_{4.Middle}$	$A_{4.High}$	$A_{5.Low}$	$A_{5.Middle}$	$A_{5.High}$
42	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
43	1	0	0	1	0	0	1	0	0	1	0	0	0.667	0.333	0
44	1	0	0	1	0	0	1	0	0	0.667	0.333	0	0.867	0.133	0
45	1	0	0	1	0	0	0.667	0.333	0	0.867	0.133	0	1	0	0
46	1	0	0	0.667	0.333	0	0.867	0.133	0	1	0	0	0.787	0.213	0
47	0.667	0.333	0	0.867	0.133	0	1	0	0	0.787	0.213	0	0.82	0.18	0
48	0.867	0.133	0	1	0	0	0.787	0.213	0	0.82	0.18	0	0.66	0.34	0
49	1	0	0	0.787	0.213	0	0.82	0.18	0	0.66	0.34	0	0.94	0.06	0
50	0.787	0.213	0	0.82	0.18	0	0.66	0.34	0	0.94	0.06	0	0.727	0.273	0
51	0.82	0.18	0	0.66	0.34	0	0.94	0.06	0	0.727	0.273	0	0.727	0.273	0
52	0.66	0.34	0	0.94	0.06	0	0.727	0.273	0	0.727	0.273	0	0.767	0.233	0
53	0.94	0.06	0	0.727	0.273	0	0.727	0.273	0	0.767	0.233	0	0.053	0.947	0
54	0.727	0.273	0	0.727	0.273	0	0.767	0.233	0	0.053	0.947	0	0	0.867	0.133
55	0.727	0.273	0	0.767	0.233	0	0.053	0.947	0	0	0.867	0.133	0	0.467	0.533
56	0.767	0.233	0	0.053	0.947	0	0	0.867	0.133	0	0.467	0.533	0	0.08	0.92
57	0.053	0.947	0	0	0.867	0.133	0	0.467	0.533	0	0.08	0.92	0	0.633	0.367
58	0	0.867	0.133	0	0.467	0.533	0	0.08	0.92	0	0.633	0.367	0.727	0.273	0
59	0	0.467	0.533	0	0.08	0.92	0	0.633	0.367	0.727	0.273	0	0.953	0.047	0
60	0	0.08	0.92	0	0.633	0.367	0.727	0.273	0	0.953	0.047	0	0.907	0.093	0
61	0	0.633	0.367	0.727	0.273	0	0.953	0.047	0	0.907	0.093	0	0.647	0.353	0
62	0.727	0.273	0	0.953	0.047	0	0.907	0.093	0	0.647	0.353	0	0.253	0.747	0
63	0.953	0.047	0	0.907	0.093	0	0.647	0.353	0	0.253	0.747	0	0.78	0.22	0
64	0.907	0.093	0	0.647	0.353	0	0.253	0.747	0	0.78	0.22	0	0.253	0.747	0
65	0.647	0.353	0	0.253	0.747	0	0.78	0.22	0	0.253	0.747	0	0	1	0
66	0.253	0.747	0	0.78	0.22	0	0.253	0.747	0	0	1	0	0	0.827	0.173
67	0.78	0.22	0	0.253	0.747	0	0	1	0	0	0.827	0.173	0	0	1
68	0.253	0.747	0	0	1	0	0	0.827	0.173	0	0	1	0	0.48	0.52
69	0	1	0	0	0.827	0.173	0	0	1	0	0.48	0.52	0.427	0.573	0
70	0	0.827	0.173	0	0	1	0	0.48	0.52	0.427	0.573	0	0	0.887	0.113
71	0	0	1	0	0.48	0.52	0.427	0.573	0	0	0.887	0.113	0.74	0.26	0
72	0	0.48	0.52	0.427	0.573	0	0	0.887	0.113	0.74	0.26	0	0.633	0.367	0
73	0.427	0.573	0	0	0.887	0.113	0.74	0.26	0	0.633	0.367	0	0	1	0
74	0	0.887	0.113	0.74	0.26	0	0.633	0.367	0	0	1	0	0.28	0.72	0
75	0.74	0.26	0	0.633	0.367	0	0	1	0	0.28	0.72	0	0.587	0.413	0
76	0.633	0.367	0	0	1	0	0.28	0.72	0	0.587	0.413	0	0.14	0.86	0
77	0	1	0	0.28	0.72	0	0.587	0.413	0	0.14	0.86	0	0.3	0.7	0
78	0.28	0.72	0	0.587	0.413	0	0.14	0.86	0	0.3	0.7	0	0.313	0.687	0
79	0.587	0.413	0	0.14	0.86	0	0.3	0.7	0	0.313	0.687	0	0.107	0.893	0
80	0.14	0.86	0	0.3	0.7	0	0.313	0.687	0	0.107	0.893	0	0.24	0.76	0
81	0.3	0.7	0	0.313	0.687	0	0.107	0.893	0	0.24	0.76	0	0.707	0.293	0

Table 9: Continued.

Count	47.595	60.227	8.178	47.275	60.547	8.178	46.802	61.02	8.178	47.262	60.56	8.178	47.775	60.047	8.178
Sp	$A_{1.Low}$	$A_{1.Middle}$	$A_{1.High}$	$A_{2.Low}$	$A_{2.Middle}$	$A_{2.High}$	$A_{3.Low}$	$A_{3.Middle}$	$A_{3.High}$	$A_{4.Low}$	$A_{4.Middle}$	$A_{4.High}$	$A_{5.Low}$	$A_{5.Middle}$	$A_{5.High}$
82	0.313	0.687	0	0.107	0.893	0	0.24	0.76	0	0.707	0.293	0	0.433	0.567	0
83	0.107	0.893	0	0.24	0.76	0	0.707	0.293	0	0.433	0.567	0	0.647	0.353	0
84	0.24	0.76	0	0.707	0.293	0	0.433	0.567	0	0.647	0.353	0	0.293	0.707	0
85	0.707	0.293	0	0.433	0.567	0	0.647	0.353	0	0.293	0.707	0	0	0.9	0.1
86	0.433	0.567	0	0.647	0.353	0	0.293	0.707	0	0	0.9	0.1	0.307	0.693	0
87	0.647	0.353	0	0.293	0.707	0	0	0.9	0.1	0.307	0.693	0	0.98	0.02	0
88	0.293	0.707	0	0	0.9	0.1	0.307	0.693	0	0.98	0.02	0	0	0.76	0.24
89	0	0.9	0.1	0.307	0.693	0	0.98	0.02	0	0	0.76	0.24	0	1	0
90	0.307	0.693	0	0.98	0.02	0	0	0.76	0.24	0	1	0	0	0.807	0.193
91	0.98	0.02	0	0	0.76	0.24	0	1	0	0	0.807	0.193	0	0.533	0.467
92	0	0.76	0.24	0	1	0	0	0.807	0.193	0	0.533	0.467	0	0.207	0.793
93	0	1	0	0	0.807	0.193	0	0.533	0.467	0	0.207	0.793	0.133	0.867	0
94	0	0.807	0.193	0	0.533	0.467	0	0.207	0.793	0.133	0.867	0	0.173	0.827	0
95	0	0.533	0.467	0	0.207	0.793	0.133	0.867	0	0.173	0.827	0	0.22	0.78	0
96	0	0.207	0.793	0.133	0.867	0	0.173	0.827	0	0.22	0.78	0	0.413	0.587	0
97	0.133	0.867	0	0.173	0.827	0	0.22	0.78	0	0.413	0.587	0	0.493	0.507	0
98	0.173	0.827	0	0.22	0.78	0	0.413	0.587	0	0.493	0.507	0	0.807	0.193	0
99	0.22	0.78	0	0.413	0.587	0	0.493	0.507	0	0.807	0.193	0	1	0	0
100	0.413	0.587	0	0.493	0.507	0	0.807	0.193	0	1	0	0	0.413	0.587	0
101	0.493	0.507	0	0.807	0.193	0	1	0	0	0.413	0.587	0	0.54	0.46	0
102	0.807	0.193	0	1	0	0	0.413	0.587	0	0.54	0.46	0	0	1	0
103	1	0	0	0.413	0.587	0	0.54	0.46	0	0	1	0	0.033	0.967	0
104	0.413	0.587	0	0.54	0.46	0	0	1	0	0.033	0.967	0	0.087	0.913	0
105	0.54	0.46	0	0	1	0	0.033	0.967	0	0.087	0.913	0	1	0	0
106	0	1	0	0.033	0.967	0	0.087	0.913	0	1	0	0	0.147	0.853	0
107	0.033	0.967	0	0.087	0.913	0	1	0	0	0.147	0.853	0	0.473	0.527	0
108	0.087	0.913	0	1	0	0	0.147	0.853	0	0.473	0.527	0	0.253	0.747	0
109	1	0	0	0.147	0.853	0	0.473	0.527	0	0.253	0.747	0	0.813	0.187	0
110	0.147	0.853	0	0.473	0.527	0	0.253	0.747	0	0.813	0.187	0	0.64	0.36	0
111	0.473	0.527	0	0.253	0.747	0	0.813	0.187	0	0.64	0.36	0	0.667	0.333	0
112	0.253	0.747	0	0.813	0.187	0	0.64	0.36	0	0.667	0.333	0	0.387	0.613	0
113	0.813	0.187	0	0.64	0.36	0	0.667	0.333	0	0.387	0.613	0	0.68	0.32	0
114	0.64	0.36	0	0.667	0.333	0	0.387	0.613	0	0.68	0.32	0	0.527	0.473	0
115	0.667	0.333	0	0.387	0.613	0	0.68	0.32	0	0.527	0.473	0	0.46	0.54	0
116	0.387	0.613	0	0.68	0.32	0	0.527	0.473	0	0.46	0.54	0	0.58	0.42	0

for analysing the change activity of OSS projects. Subject systems were selected from public repositories but selection is biased towards projects with valid *git* repositories.

External validity concerns the generalization of the findings. In the future, we would like to provide more generalized results by considering higher number of OSS projects.

Reliability validity concerns the possibility of replication of the study. The subject systems are available in the public domain. We have attempted to put all the necessary details of the experiment process in the paper.

8. Conclusion and Future Work

The commit activity data available in the development repository of open source software can be used to analyse the evolution of OSS projects as each commit is directly related to the development activity such as code deletion, addition, modification, comments, and file addition. In this study, the Eclipse CDT commits data is analysed to find the regularity and trend in the commit data. The fuzzy data mining algorithm for time series data is used to generate the association rules from the dataset. The dataset is divided into two subsets (training and remaining dataset) to evaluate the pattern of evolution of the Eclipse CDT.

After applying and validating the generated association rule from training dataset, it is found that the rates at which commits performed in training dataset and remaining dataset are different. This thing is again verified by generating the association rules from the remaining set. The generated rules from remaining dataset specify that the data in the considered remaining dataset is more towards lower range of commits performed. This thing validates the applicability of the rules generated from the training dataset.

These association rules indicate that the overall commits in the Eclipse CDT are towards *middle* range except the variation found near the end, where there is a high probability that commits lie in the *lower* range. The continuous availability and existence of commits in the repository of the Eclipse CDT illustrate that the development or evolution of Eclipse CDT is active with most of the commits per month that lie in the *middle* range and at the end lie near to the *lower* range. In the future, we want to consider any prediction algorithm along with the concept of fuzzy data mining algorithm for time series data to give a prediction about the number of commits to be performed in the particular month, although there are other various factors that need to be considered on which the number of commits depends.

Appendix

See Tables 8 and 9.

Competing Interests

The authors declare that they have no competing interests.

References

[1] M. W. Godfrey and Q. Tu, "Evolution in open source software: a case study," in *Proceedings of the IEEE Interantional Conference on Software Maintenance (ICMS '00)*, pp. 131–142, October 2000.

[2] H. Zhang and S. Kim, "Monitoring software quality evolution for defects," *IEEE Software*, vol. 27, no. 4, pp. 58–64, 2010.

[3] Y. Fang and D. Neufeld, "Understanding sustained participation in open source software projects," *Journal of Management Information Systems*, vol. 25, no. 4, pp. 9–50, 2008.

[4] "GIT," July 2015, http://git-scm.com/.

[5] R. Sen, S. S. Singh, and S. Borle, "Open source software success: measures and analysis," *Decision Support Systems*, vol. 52, no. 2, pp. 364–372, 2012.

[6] R. McCleary, R. A. Hay, E. E. Meidinger, and D. McDowall, *Applied Time Series Analysis for the Social Sciences*, Sage, Beverly Hills, Calif, USA, 1980.

[7] R. Grewal, G. L. Lilien, and G. Mallapragada, "Location, location, location: how network embeddedness affects project success in open source systems," *Management Science*, vol. 52, no. 7, pp. 1043–1056, 2006.

[8] I. Herraiz, J. M. Gonzalez-Barahona, G. Robles, and D. M. German, "On the prediction of the evolution of libre software projects," in *Proceedings of the 23rd International Conference on Software Maintenance (ICSM '07)*, pp. 405–414, Paris, France, October 2007.

[9] C. F. Kemerer and S. Slaughter, "An empirical approach to studying software evolution," *IEEE Transactions on Software Engineering*, vol. 25, no. 4, pp. 493–509, 1999.

[10] A. Mockus and L. G. Votta, "Identifying reasons for software Changes using historic databases," in *Proceedings of the International Conference on Software Maintenance (ICSM '00)*, pp. 120–130, IEEE, San Jose, Calif, USA, 2000.

[11] C.-H. Chen, T.-P. Hong, and V. S. Tseng, "Fuzzy data mining for time-series data," *Applied Soft Computing Journal*, vol. 12, no. 1, pp. 536–542, 2012.

[12] C. C. H. Yuen, "An empirical approach to the study of errors in large software under maintenance," in *Proceedings of the IEEE International Conference on Software Maintenance (ICSM '85)*, pp. 96–105, 1985.

[13] C. C. H. Yuen, "A statistical rationale for evolution dynamics concepts," in *Proceedings of the IEEE International Conference on Software Maintenance (ICSM '87)*, pp. 156–164, September 1987.

[14] C. C. H. Yuen, "On analytic maintenance process data at the global and the detailed levels: a case study," in *Proceedings of the International Conference on Software Maintenance (ICSM '88)*, pp. 248–255, IEEE, Scottsdale, Ariz, USA, 1988.

[15] M. Goulão, N. Fonte, M. Wermelinger, and F. B. Abreu, "Software evolution prediction using seasonal time analysis: a comparative study," in *Proceedings of the 16th European Conference on Software Maintenance and Reengineering (CSMR '12)*, pp. 213–222, IEEE, Szeged, Hungary, March 2012.

[16] B. Kenmei, G. Antoniol, and M. Di Penta, "Trend analysis and issue prediction in large-scale open source systems," in *Proceedings of the 12th IEEE European Conference on Software Maintenance and Reengineering (CSMR '08)*, pp. 73–82, Athens, Greece, April 2008.

[17] F. Caprio, G. Casazza, U. Penta, M. D. Penta, and U. Villano, "Measuring and predicting the Linux kernel evolution," in *Proceedings of the International Workshop of Empirical Studies on Software Maintenance*, Florence, Italy, November 2001.

[18] E. Fuentetaja and D. J. Bagert, "Software evolution from a time-series perspective," in *Proceedings of the IEEE International Conference on Software Maintenance*, pp. 226–229, October 2002.

[19] M. Kläs, F. Elberzhager, J. Münch, K. Hartjes, and O. Von Graevemeyer, "Transparent combination of expert and measurement data for defect prediction—An Industrial Case Study," in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE '10)*, pp. 119–128, Cape Town, South Africa, May 2010.

[20] U. Raja, D. P. Hale, and J. E. Hale, "Modeling software evolution defects: a time series approach," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 21, no. 1, pp. 49–71, 2009.

[21] G. Antoniol, G. Casazza, M. D. Penta, and E. Merlo, "Modeling clones evolution through time series," in *Proceedings of the IEEE International Conference on Software Maintenance (ICSM '01)*, pp. 273–280, IEEE Computer Society, Florence, Italy, November 2001.

[22] L. Yu, "Indirectly predicting the maintenance effort of open-source software," *Journal of Software Maintenance and Evolution*, vol. 18, no. 5, pp. 311–332, 2006.

[23] A. Amin, L. Grunske, and A. Colman, "An approach to software reliability prediction based on time series modeling," *Journal of Systems and Software*, vol. 86, no. 7, pp. 1923–1932, 2013.

[24] Forecasting Model, http://people.duke.edu/~rnau/411fcst.htm.

[25] T.-P. Hong, C.-S. Kuo, and C.-C. Chi, "A fuzzy data mining algorithm for quantitative values," in *Proceedings of the 3rd International Conference on Knowledge-Based Intelligent Information Engineering Systems (KES '99)*, pp. 480–483, September 1999.

[26] H. Suresh and K. Raimond, "Mining association rules from time series data using hybrid approaches," *Proceedings of International Journal of Computational Engineering Research*, vol. 3, no. 3, pp. 181–189, 2013.

[27] "GIT HUB," July 2015, https://github.com/.

[28] "Eclipse", "Where did Eclipse come from?", Eclipse Wiki.

[29] "Eclipse," 2015, https://eclipse.org/cdt/.

[30] "Eclipse," July 2015, http://www.eclipse.org/cdt/doc/overview/#/3.

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

[32] J. Han and M. Kamber, *Data Mining Concepts and Techniques*, Morgan Kaufman Academic Press, 2001.

[33] R. Agrawal, "Mining association rules between sets of items in large databases," in *Proceedings of the ACM SIGMOD Conference*, pp. 207–216, Washington, DC, USA, May 1993.

Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2016, Article ID 6134736, 19 pages
http://dx.doi.org/10.1155/2016/6134736

Research Article

An Improved Fuzzy Based Missing Value Estimation in DNA Microarray Validated by Gene Ranking

Sujay Saha,[1] Anupam Ghosh,[2] Dibyendu Bikash Seal,[3] and Kashi Nath Dey[4]

[1]*Department of Computer Science and Engineering, Heritage Institute of Technology, Kolkata 700107, India*
[2]*Department of Computer Science and Engineering, Netaji Subhash Engineering College, Kolkata 700152, India*
[3]*Department of Computer Science and Engineering, University of Engineering & Management, Kolkata 700156, India*
[4]*Department of Computer Science and Engineering, University of Calcutta, Kolkata 700098, India*

Correspondence should be addressed to Sujay Saha; sujay.saha@heritageit.edu

Received 22 March 2016; Accepted 16 June 2016

Academic Editor: Gözde Ulutagay

Most of the gene expression data analysis algorithms require the entire gene expression matrix without any missing values. Hence, it is necessary to devise methods which would impute missing data values accurately. There exist a number of imputation algorithms to estimate those missing values. This work starts with a microarray dataset containing multiple missing values. We first apply the modified version of the fuzzy theory based existing method LRFDVImpute to impute multiple missing values of time series gene expression data and then validate the result of imputation by genetic algorithm (GA) based gene ranking methodology along with some regular statistical validation techniques, like RMSE method. Gene ranking, as far as our knowledge, has not been used yet to validate the result of missing value estimation. Firstly, the proposed method has been tested on the very popular Spellman dataset and results show that error margins have been drastically reduced compared to some previous works, which indirectly validates the statistical significance of the proposed method. Then it has been applied on four other 2-class benchmark datasets, like Colorectal Cancer tumours dataset (GDS4382), Breast Cancer dataset (GSE349-350), Prostate Cancer dataset, and DLBCL-FL (Leukaemia) for both missing value estimation and ranking the genes, and the results show that the proposed method can reach 100% classification accuracy with very few dominant genes, which indirectly validates the biological significance of the proposed method.

1. Introduction

Microarray expression analysis is a widely used technique for profiling mRNA expression. The mRNA carries genetic information from DNA to the ribosome, where they specify the amino acid sequence of the protein products of gene expression. Microarray datasets often contain missing values which may occur due to various reasons including imperfections in data preparation steps (e.g., poor hybridization and chip contamination by dust and scratches) that create erroneous and low-quality values, which are usually discarded and referred to as missing. It is common for gene expression data to contain at least 5% missing values [1]. Most of the microarray data analysis algorithms, such as gene clustering, disease (experiment) classification, and gene network design, require the complete information, that is, the entire gene expression matrix without any missing values. Hence, different imputation techniques should be used which would accurately impute multiple missing data values. Numerous imputation algorithms have been proposed to estimate the missing values. At first, we have applied modified version of our existing imputation technique LRFDVImpute [2] that first finds a subset of similar genes using the fuzzy difference vector (FDV) algorithm used in [3] where gene expression profiles have been considered as continuous time series curves and then use linear regression on the subset to estimate the missing value. We have considered estimating only those genes with one, two, or three missing values since these genes constitute 5–10% of the entire dataset. Absolute error has been calculated from the difference between the

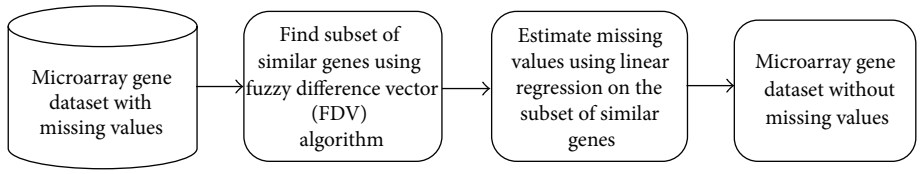

Figure 1: Workflow of the proposed missing value estimation technique.

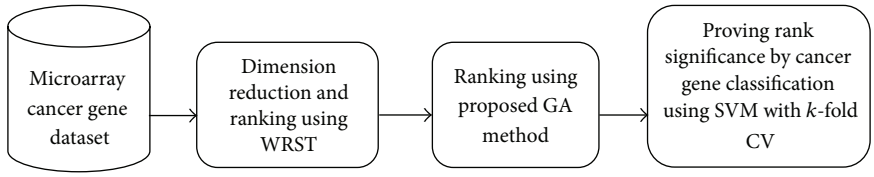

Figure 2: Workflow of the proposed gene ranking technique.

original value and the estimated value. Root Mean Square Error (RMSE) of those absolute errors is then determined.

The workflow for the first phase has been shown in Figure 1.

After that we rank those genes to find the top ranked genes [4]. We have used a hypothesis test, Wilcoxon rank sum test [5], to sort the features (genes) and rank them in order of their p values and select top n genes from them, thereby reducing the dimensionality, where n is the population size that has been used later for GA. The reduced set of genes has then been ranked by our GA method. The two ranks, one by Wilcoxon method and the other by our GA method, are then compared. The top m genes (value of m defined by the user) selected by our method are then used for classification using support vector machine (SVM) classifiers. The performance of classification justifies the efficiency of the ranking method used. Figure 2 shows the workflow for this phase.

Once this is done, we then forcibly make some cells missing in the top ranked genes and again estimate them using the same missing value estimation technique. Finally, we rank them once more to find the top ranked genes. Results show that most of the top ranked genes remain the same, which validates the proposed missing value estimation technique biologically as far as the estimation is concerned.

2. Present State of the Art

As discussed earlier, various statistical and analytical methods used for gene expression analysis are not robust to missing values and require the complete gene expression matrix for providing accurate results. Hence, it is necessary to devise accurate methods which would impute data values when they are missing. Many imputation methods have been proposed. The earliest method, named as row averaging or filling with zeroes, used to fill in the gaps for the missing values in gene dataset with zeroes or with the row average.

KNNImpute method proposed in [1] selects genes with expression profiles similar to the gene of interest to impute missing values. After experimenting with a number of metrics to calculate the gene similarity, such as Pearson correlation, Euclidian distance, and variance minimization, it was found that Euclidian distance was a sufficiently accurate norm.

The SVDImpute method, proposed in [1], uses Singular Value Decomposition of matrices to estimate the missing values of a DNA microarray. This method works by decomposing the gene data matrix into a set of mutually orthogonal expression patterns that can be linearly combined to approximate the expression of all genes in the dataset. These patterns, which in this case are identical to the principle components of the gene expression matrix, are further referred to as eigengenes [6, 7].

Another method named as LLSImpute [8] represents a target gene with missing values as a linear combination of similar genes. The similar genes are chosen by k-nearest neighbours or k coherent genes that have large absolute values of correlation coefficients followed by least square regression and estimation.

BPCAImpute method, proposed in [9], uses a Bayesian estimation algorithm to predict missing values. BPCA suggests using the number of samples minus 1 as the number of principal axes. Since BPCA uses an EM-like repetitive algorithm to estimate missing values, it needs intensive computations to impute missing values.

Another algorithm for time series gene expression analysis is presented in [10] that permits the principled estimation of unobserved time points, clustering, and dataset alignment. Each expression profile is modelled as a cubic spline (piecewise polynomial) that is estimated from the observed data and every time point influences the overall smooth expression curve. The alignment algorithm uses the

same spline representation of continuous time series gene expression profiles.

FDVImpute method, proposed in [11], incorporates some fuzziness to estimate the missing value of a DNA microarray. The first step selects nearest (most similar) genes of the target gene (whose some component is missing) using fuzzy difference vector algorithm. Then the missing cell is estimated by using least square fit on the selected genes in the second step.

FDVSplineImpute, presented in [3], takes into account the time series nature of gene expression data and permits the estimation of missing observations using B-splines of similar genes from fuzzy difference vectors.

Another method, LRFDVImpute, proposed in [2], estimates multiple missing observations by first finding the most similar genes of the target gene and then applying the linear regression on those similar genes. This approach works in two stages. At the first stage, it estimates the real missing cells of SPELLMAN_COMBINED dataset and at the later stage, it makes some cells miss forcefully of the same dataset and then using the estimated results from the first step, this approach estimates those missed cells using the same approach used earlier. Absolute error has been calculated from the difference between the original value and the estimated value. Root Mean Square Error (RMSE) of those absolute errors is then determined.

Extracting relevant information from microarray data is also difficult because of the inherent characteristics of the datasets, where there are the thousands of variables (genes) and very few numbers of samples. Finding out the set of significant genes or, in other words, the most differentially expressed genes, by studying data from tissues affected or unaffected by cancer cells, is an important task. This problem can be termed as gene selection. Several techniques have been used to rank genes and find out the most significant ones.

In [12], the algorithm used discriminant partial least squares (DPLS) and fuzzy clustering methods to interpret the gene expression patterns of acute leukemia and identify leukemia subtypes.

In [13], the proposed method used Mann-Whitney test and k-sample Kruskal-Wallis ANOVA test to rank genes. Dimension reduction was done using k-means clustering and PCA and classification performed using ANN trained during 8-fold cross-validation with recursive feature elimination (RFE) and leave-one-out testing.

In [14], the algorithm proposed a gene selection method based on Wilcoxon rank sum test and SVM. Wilcoxon rank sum test was used to select a subset of genes and then each selected gene is trained and tested using SVM classifier with linear kernel separately, and genes with high testing accuracy rates were chosen to form the final reduced gene subset. Classification was performed on two datasets: Breast Cancer [15] and ALL/AML Leukemia [16] using leave-one-out cross-validation (LOOCV).

A hybrid GA/SVM approach is proposed for gene selection in [17], where a fuzzy logic based preprocessing tool is used to reduce dimensionality, GA for finding out the most frequent genes, and a SVM classifier used for classification. Experiments were performed on two well-known cancer datasets, Leukemia [16] and Colon [18], and results were compared with six other methods.

A multiobjective genetic approach is proposed in [19] for simultaneous clustering and gene ranking where a method to simultaneously optimize the feature ranking and clustering has been used. NSGA-II (Nondominated Sorting Genetic Algorithm-II) [20] has been used as a multiobjective evolutionary algorithm to optimize the chromosomes.

In [21], the proposed algorithm uses feature selection method based on genetic algorithms (GAs) and classification methods focusing on constructive neural networks (CNNs), C-Mantec. Several comparison results on six public cancer databases are provided using other feature selection strategy (Stepwise Forward Selection method) and different classification techniques (LDA, SVM, and Naive Bayes).

A PSO based graph theoretic approach, proposed in [22], is used for identifying the nonredundant gene markers from microarray gene expression data. The microarray data is first converted into a weighted undirected complete feature graph where the nodes represent the genes having gene's relevance as node weights and the edges are weighted in order of correlation among the genes. The densest subgraph having minimum average edge weight (similarity) and maximum average node weight (relevance) is then identified from the original feature graph. Binary particle swarm optimization is then applied for minimizing the average edge weight (correlation) and maximizing the average node weight (gene relevance) through a single objective function.

A web based tool DWFS, proposed in [23], is used to select significant features for a variety of problems efficiently. The search strategy is implemented using Parallel Genetic Algorithm. DWFS also applies various filtering methods as a preprocessing step in the feature selection process. It also uses three classifiers, like KNN classifier, Naive Bayes Classifier, and the combination of these two. Experiments using datasets taken from different biomedical applications show the efficiency of DWFS and lead to a significant reduction of the number of features without sacrificing performance as compared to several widely used existing methods.

3. Proposed Method

3.1. Missing Value Estimation Using Linear Regression. This phase of the work modifies an existing method LRFDVImpute for estimating missing values present in the microarray dataset using linear regression. Earlier version of LRFDVImpute inserts the newly estimated gene into the training data after estimation of each target gene. In this way, the newly estimated gene is taken into consideration while estimating the next target gene. This process has the risk of increasing the error while estimating the subsequent genes since the error term is cumulatively multiplied. To overcome this problem, modified LRFDVImpute does not add the target gene to the training data after it has been estimated. This way, the training gene set size remains constant and with increasing membership values of θ, the size of training data reduces. The effects of modifications have been studied and results are shown in the experimental results section. In our problem,

the genes with missing values in the $(p \times q)$ (p is the number of genes and q is the number of samples) dataset are to be estimated. The method of finding a similar gene as used in [3] using fuzzy difference vector (FDV) algorithm is described below.

Target Row/Testing Data. The row whose missing value is being estimated: a target row may have multiple missing values but in a single run, a single value is estimated.

Similar Rows/Training Data. The rows that are similar to the target row: in this case only those rows are selected that have no missing values. Before applying the similarity measures all the columns from the complete matrix are removed that correspond to missing values in target row.

Let $g_1, g_2, \ldots, g_p$ be the set of genes in the dataset. Let ith g_i be the target gene, that is, the gene with m missing values. We remove the columns having missing values from the entire dataset. Let the resultant matrix contain $(q - m)$ columns. Each target gene i is compared with each of the similar rows in the dataset. For the ith gene g_i, the difference vector V_i of g_i is calculated as follows:

$$\text{DifferenceTable}_{i,j} = g_i(j) - g_i(j+1), \quad 1 \le j \le q - m - 1. \tag{1}$$

Once the difference vectors are calculated for each of the target rows and the similar rows, say DifferenceTable_1 (for target row) and DifferenceTable_2 (for similar row), we then calculate Membership(i) to obtain the number of matches between difference vectors DifferenceTable_1 and DifferenceTable_2 for each target gene g_i. A match in the jth component of the vectors DifferenceTable_1 and DifferenceTable_2 is determined by whether the signs of $\text{DifferenceTable}_1(i, j)$ and $\text{DifferenceTable}_2(i, j)$ are the same or not. Membership(i) defines the degree of match between the distribution of the target gene and the similar gene. We then define a membership grade for g_i as follows:

$$\text{memgrade}(i) = \frac{\text{Membership}(i)}{(q - m - 1)}. \tag{2}$$

The genes in the training data that have a membership value greater than a chosen membership grade θ are considered to be a part of the similar genes.

The steps for estimation can be summarized below:

(1) Load the dataset with missing values.

(2) Calculate the missing number of columns for each gene and start with the first row with the least number of missing values (for our dataset it is 1).

(3) Compute the corresponding membership grade for the target gene from the training data using the FDV algorithm as shown above.

(4) Estimate the missing value using linear regression.

(5) Obtain coefficients of the regression from the linear model object lmObj.

(6) Add a bias of 1 at the beginning of the target row to allow for the bias parameter.

(7) Perform a vector multiplication between the modified target row and the coefficients of regression and add the obtained vector's elements together to get the estimated value.

(8) Replace the missing value with the estimated value.

(9) Go to step (2) and repeat the above steps to fill the missing values unless the mentioned "least number of missing values" in step (2) is less than or equal to 3.

Although we mentioned here that we go on filling the missing value till a point, it is not true. In between we stop this filling in process to do assessment of our algorithm.

After we have filled in all the missing values corresponding to rows with single missing values we select a particular collection of row-column positions corresponding to rows that did not have missing values initially and deliberately treat the values at these positions as missing and use the exact same process to estimate the values.

The same collection of row-column positions are again used when the algorithm has filled up all the rows up to two missing vales and then when it has filled up missing values existing in rows with up to three missing values.

3.2. Gene Ranking Using Genetic Algorithm. In phase 2 of the proposed work, the result of the missing value estimation procedure carried out in phase 1 is biologically validated by ranking the genes using GA. Since a characteristic of gene expression microarray data is that the number of variables (genes) far exceeds the number of samples n, we must reduce its dimension. Executing GA on the original dataset is quite impractical and time consuming. As a preprocessing step, we have reduced the dimension using Wilcoxon rank sum test.

3.2.1. Dimension Reduction Using Wilcoxon Rank Sum Test (WRST). The inputs to the Wilcoxon rank sum test function are the two gene sets, the diseased set and the normal set, both of which have individually undergone the missing value estimation procedure (if there was any missing value). The two gene sets may have different number of samples. Let us consider that the diseased set is a $(p \times q_1)$ sized gene expression data, where p is the number of genes and q_1 is the number of samples, and the normal set has a size $(p \times q_2)$, where q_2 is the number of samples. The Wilcoxon rank sum function processes the two datasets in order to find out for which genes the null hypothesis is accepted or rejected. It returns two values, p value and h-value, as discussed earlier. The null hypothesis for our problem is that the genes are not differentially expressed; that is, either all the samples have come from diseased patients or they have come from normal patients. The alternative hypothesis can be that genes are differentially expressed. We record the p values and h-values for each gene.

In the next step, we consider only those genes for which the alternative hypothesis holds ($h = 1$) at the significance level alpha and sort the genes according to the p values thereby ranking the genes. We then select the topmost k

genes, where k is the population size that has been used for GA later. Thus, we have two reduced populations, one representing diseased and the other representing normal tissues. Let $(X_{ij})_{p\times q_1}$ be the diseased set, where p is the reduced set of genes and q_1 is the number of samples, respectively, and let $(Y_{ij})_{p\times q_2}$ be the normal set, where q_2 is the number of samples.

3.2.2. Chromosome Representation and Initial Population for GA. The reduced gene sets $(X_{ij})_{p\times q_1}$ and $(Y_{ij})_{p\times q_2}$ serve as the initial population for the genetic algorithm step. They contain pop_size number of genes which is preselected by the user. We use real value encoding to represent each chromosome; that is, X_{ij} and Y_{ij} are the measurements recorded for the ith gene and jth sample for each population, respectively.

3.2.3. Fitness Calculation. The fitness for each gene in the reduced gene sets is again calculated by a method similar to that used in [14] where gene expression profiles have been considered as continuous time series curves.

In our problem, we have two populations, one for the diseased tissues and the other for the normal tissues. The two populations contain the same number of genes p but may have different number of samples. In that case, we consider the minimum of the two and extract the same number of samples from each set.

Let $g_1, g_2, \ldots, g_p$ be the reduced set of genes in each population. If $q = \min(q_1, q_2)$, then for each population, the difference vector V_i of g_i is calculated using (1). Once the difference vectors are calculated for each of the two populations, say DifferenceTable_1 (for diseased) and DifferenceTable_2 (for normal), the number of matches between the difference vectors Membership(i) and the membership grade for g_i is computed using (2).

The fitness of gene g_i is the reciprocal of memgrade(i) and is calculated as

$$\text{fit}(i) = \frac{1}{\text{memgrade}(i)}. \tag{3}$$

This signifies that the more similar the distributions of gene g_i in the two populations are, the less differentially expressed the gene is, and vice versa. Thus, a fitter gene will have different distributions in the two populations. We then rank the genes in order of their fitness.

3.2.4. Elitism. We have used an elitist version of GA where the best chromosomes are carried forward to the next generation unchanged; that is, the crossover and mutation operators are not applied on the best chromosomes. This technique ensures faster convergence of the process by keeping track of the best solutions.

3.2.5. Selection. For selection, we have used a roulette wheel technique where genes are selected based on their relative fitness values. The better the chromosomes are, the more chances to be selected they have. Let count be the number of elite children. We construct a roulette wheel as follows [22]:

(i) Calculate the fitness value fit(i) for each chromosome g_i, count + 1 ≤ i ≤ p.

(ii) Find the total fitness of the population $F = \sum_{i=\text{count}+1}^{\text{pop\_size}} \text{fit}(g_i)$.

(iii) Calculate the probability of selection p_i for each chromosome g_i, count + 1 ≤ i ≤ p:

$$p_i = \frac{\text{fit}(g_i)}{F}. \tag{4}$$

(iv) Calculate a cumulative probability c_i for each chromosome g_i, count + 1 ≤ i ≤ p:

$$c_i = \sum_{j=1}^{i} p_j. \tag{5}$$

We now spin the wheel (pop_size − count) times and select a single chromosome as follows:

(i) Generate a random number (float) r between 0 and 1.

(ii) If $r < c_1$, we select the first chromosome g_1; otherwise, select the ith chromosome g_i ($2 \le i \le$ pop_size − count) such that $c_{i-1} < r \le c_i$.

Some chromosomes get selected more than once. According to *Schema Theorem* [24], the best chromosomes get more copies, the average stay even, and the worst die off.

3.2.6. Crossover. For crossover, we proceed as follows.

For each chromosome g_i in the population,

(i) generate a random number (float) r between 0 and 1,

(ii) if $r < p_{\text{cross}}$ (crossover probability), we select the given chromosome for crossover.

We have used single point crossover where the crossover site is also generated randomly in the range $[1 \cdots q-1]$, where q is the number of samples. Thus after crossover, a pair of parent chromosomes generates a pair of offspring chromosomes [25]. The new population obtained after crossover contains the new generation produced by crossover as well as the elite children that did not undergo crossover. This new population is used in the mutation process.

3.2.7. Mutation. A nonuniform mutation operator as proposed in literature [25] has been used here. The new operator is defined as follows:

(i) A random experiment is carried out which produces an outcome which is either 0 or 1.

(ii) Another random number pos is generated in the range $[1 \cdots q-1]$, where q is the number of samples, to select the mutation site.

(iii) Let $g_i^t = [g_i(1), g_i(2), \ldots, g_i(j), \ldots, g_i(q)]$, $1 \le j \le q$, be the chromosome, and let $g_i(j)$ be selected for mutation. Domain of g_i is [lb, ub]; the resultant vector $g_i^{t+1} = [g_i(1), g_i(2), \ldots, g_i(j)', \ldots, g_i(q)]$:

Table 1: Characteristic of Spellman dataset.

Dataset	Start	End	Sampling	Complete genes
alpha	0 m	119 m	Every 7 m	4489
cdc15	10 m	290 m	Every 20 m for 1 hr, 10 m for 3 hr, and 20 m for final hr	4381
cdc28	0 m	160 m	Every 10 m	1383
elu	0 m	390 m	Every 30 m	5766

Yeast *Saccharomyces cerevisiae* dataset of Spellman et al. [26].
Source: http://genome-www.stanford.edu/cellcycle/data/rawdata/combined.txt.
Organism: yeast.

$$g_i(j)' = \begin{cases} g_i(j) + \Delta(t, \text{ub} - g_i(j)), & \text{if outcome is 0} \\ g_i(j) - \Delta(t, g_i(j) - \text{lb}), & \text{if outcome is 1,} \end{cases} \quad (6)$$

where t is the generation number and the function $\Delta(t, y)$ returns a value in the range $[0 \cdots y]$ such that the probability of $\Delta(t, y)$ being close to 0 increases as t increases. This property causes this operator to search the space uniformly initially (when t is small) and very locally at later stages.

$\Delta(t, y)$ is calculated as

$$\Delta(t, y) = y\left(1 - r^{(1-t/T)^{\beta}}\right), \quad (7)$$

where r is a random number in the range $[0 \cdots 1]$, T is the maximum number of generations preselected by the user, and β is a system parameter determining the degree of uniformity. We have used $\beta = 2$ for our experiment.

The entire genetic transformation has been performed on one population with respect to the other. We made the diseased gene set to undergo genetic transformation while fitness evaluation has been made with respect to the normal gene set. The opposite transformation will produce similar results.

Once the genetic transformations are done, we obtain a final population set (here, genetically transformed diseased gene set) which have been ranked in order of their fitness. We compare the two ranks, one by the Wilcoxon method and the other by our GA method. A threshold of ±2 has been considered while comparing the two ranks. Results show that there is a good percentage of matches in the two ranks. Moreover, we find out the top ranked genes produced by both methods and the significant genes produced by the two methods are also similar. This also validates the result of the missing value estimation method carried out in phase 1.

3.3. Gene Classification Using SVM. In order to prove the significance of ranking by our GA method, we perform classification. The top ranked n genes, n' $\{5, 10, 15, 20, 25\}$, ranked by our GA method are used for the purpose. We use k-fold LOO cross-validations, where k is varied from one dataset to another depending on the number of samples. For cross-validation, we have divided our dataset into two sets, a training set and a testing set, in 80 : 20 ratio. The reason behind taking this ratio is that 80 : 20 is a commonly occurring ratio, which is often referred to as Pareto Principle. So, if there are n samples in the training set and $m - n$ samples in the test set, where m is the total number of samples, the training set is divided into k equal sized subsets. Of the k subsets, one subset is retained for validation and the remaining $k-1$ subsets are used as training data. Thus, k SVM classifiers with linear kernel are trained using the n training subsets. The classification accuracy rates are recorded and the classifier with the best accuracy rate is used to test the $m - n$ samples.

4. Experimental Results

4.1. Datasets Used. The missing value estimation part of the proposed modified LRFDVImpute technique has been evaluated on the publicly available yeast cell cycle time series dataset from Spellman et al. [26] described in Table 1.

After the experiments on Spellman dataset are done, the combined gene ranking and classification portion of the proposed method are evaluated on four publicly available datasets: Colorectal Cancer tumours dataset (GDS4382), Breast Cancer dataset (GSE349-350), Prostate Cancer dataset, and Leukaemia Cancer dataset (DLBCL-FL).

4.2. Platform Used. All algorithms have been implemented using MATLAB R2013a in Windows 8.1.

4.3. Results

4.3.1. Results of Missing Value Estimation Part. We perform the initial estimation using modified version of LRFDVImpute with a membership grade $\theta = 0.55$. After the initial estimation is over, we forcibly treat cells at specified locations as missing and estimate them using different membership values of θ and both earlier and modified versions. This has been carried out only once, after estimating rows with single missing values and the corresponding RMSE values have been recorded. We have performed our experiments only on alpha, cdc15, and elu data of Spellman dataset. The number

Table 2: Results for missing value estimation algorithm (Spellman, alpha).

RMSE\θ	0.4	0.45	0.5	0.55	0.6	0.65	0.7
Original LRFDVImpute	0.012405344	0.012488181	0.012562782	0.012562782	0.012690904	0.012197374	0.012638865
Modified LRFDVImpute	0.012439936	0.012439366	0.012389872	0.012389872	0.012645466	0.011988263	0.013268721

Table 3: Results for missing value estimation algorithm (Spellman, cdc15).

RMSE\θ	0.4	0.45	0.5	0.55	0.6	0.65	0.7
Original LRFDVImpute	0.016832094	0.016760968	0.016706119	0.016682418	0.016768837	0.016733642	0.049482242
Modified LRFDVImpute	0.016781257	0.016710318	0.016736613	0.016723349	0.016637753	0.017023671	0.057225615

Table 4: Results for missing value estimation algorithm (Spellman, elu).

RMSE\θ	0.4	0.45	0.5	0.55	0.6	0.65	0.7
Original LRFDVImpute	0	0	0	0	0	0	0
Modified LRFDVImpute	0	0	0	0	0	0	0

Table 5: Performance comparison with other existing methods.

Dataset	SVDImpute	LLSImpute	FDVLLSImpute	FDVSPLINEImpute	FDVLRImpute with $\theta = 0.55$	
					Original LRFDVImpute	Modified LRFDVImpute
alpha	0.03395	0.07853	0.096	0.063	0.012562782	0.012389872
cdc15	0.05055	0.1208	0.258	0.127	0.016682418	0.016723349
elu	0.01585	0.0033	0.044	.019	0	0

of missing values is too large for cdc28; that is why we ignore that segment. The results for the alpha, cdc15, and elu datasets using both methods are shown in Tables 2–4. Figures 3–5 show the corresponding plots of RMSE versus membership grade θ for each of the four datasets.

Table 5 compares the performance of both versions of LRFDVImpute method to that of some other existing methods, like SVDImpute, LLSImpute, FDVLLSImpute, FDVSPLINEImpute, and so forth, and the results show that modified version of LRFDVImpute outperforms the other existing methods as far as RMSE value is concerned.

4.3.2. Combined Results. We test the significance of our proposed missing value estimation technique using the gene ranking method. We have not found any state-of-the-art work on gene ranking so far where Spellman dataset is used. That is why we use four more publicly available real-life gene expression datasets, like Colorectal Cancer dataset (GDS4382), Breast Cancer dataset (GSE349-350), Prostate Cancer dataset, and Leukaemia Cancer dataset (DLBCL-FL) [4, 27–32], to perform steps such as missing values estimation and gene ranking and analyze the results. We start with the microarray dataset containing missing values and apply our proposed missing value estimation technique to estimate the genes with missing values (if any). We rank them using proposed gene ranking method and find the top ranked genes. We then forcibly insert missing values in the top ranked genes and again estimate them using the same missing value estimation technique. Finally, we rank them once more to find the top ranked genes. Results show that most of the top ranked genes remain the same, which implies that the proposed missing value estimation technique has been accurate in estimating the unknown values. We have normalized most of the datasets using z-score normalization method in order to bring the data values to a common scale.

Tables 6, 8, 10, and 13 show the estimated values for the four datasets, Tables 7, 9, 11, and 14 show the common gene indices before and after the estimation, and Tables 12 and 15 compare the performance of the proposed approach with two state-of-the-art methods [22, 23] for Prostate and Leukaemia dataset on the basis of accuracy, sensitivity, specificity, $F1$-score, and G-mean metrics. We have found that Prostate and Leukaemia are the common dataset on which both the existing methods have done their experiments. The results show that the proposed gene ranking approach performs far better compared to those existing approaches, where one is a PSO based graph theoretic approach [22] and the other is a web based tool DWFS, which uses KNN and NBC classifiers [23] as far as those metrics are concerned.

Table 6: Estimated values for Colorectal Cancer GDS4382.

Colorectal Cancer GDS4382							
Cancer				Normal			
Missing values inserted		Old_value	Estimated value with mem = 0.55	Missing values inserted		Old_value	Estimated value with mem = 0.55
At row	At column			At row	At column		
714	12	0.677252397	0.703080766	714	12	0.140108496	−0.163286572
1245	5	1.56686079	1.755118642	1245	5	1.212949296	1.148663192
1578	3	0.787510895	1.003134829	1578	3	0.256998387	0.22381735
1763	11	1.024714768	0.737414862	1763	11	0.20681001	0.056394235
2792	9	−0.861395162	−0.865043164	2792	9	−1.064597763	−1.005338727
4025	1	0.781326343	1.137532185	4025	1	−0.31562626	−0.297767551
4134	15	0.892338308	0.958542974	4134	15	0.329755342	0.247541595
5082	2	−0.006360431	0.32763543	5082	2	−0.480000425	−0.370717546
8426	13	1.210288879	0.790345743	8426	13	−0.082823022	0.157094029
9979	6	2.932068401	2.953449691	9979	6	2.387826603	2.246418246
10083	11	1.369142269	1.258410425	10083	11	1.915272483	1.809541605
10145	10	1.940467541	1.852338203	10145	10	2.750210569	2.834220655
10208	3	1.868505436	1.778084682	10208	3	2.224446282	2.122262802
10280	6	1.424951861	1.521769713	10280	6	0.98680627	1.139738659
10323	1	0.895845032	0.963530648	10323	1	0.378278185	0.376819137
10725	14	0.562534293	0.903674258	10725	14	1.263242627	1.285858157
10789	4	1.582210172	1.834961289	10789	4	0.507667085	0.540477481
10855	10	3.17769843	3.162360832	10855	10	2.562212194	2.546399932
11050	3	2.676634486	2.657786027	11050	3	1.872943693	1.894214081
11055	8	2.112261939	2.105431748	11055	8	1.680697142	1.713521495
11100	16	2.522783399	2.455429314	11100	16	3.208462284	3.064140274
11465	1	2.454481056	2.18211664	11465	1	1.232460868	1.356633277
11485	12	−0.701537989	−0.49772711	11485	12	0.609015246	0.316407836
11650	6	1.470662615	1.35686403	11650	6	1.940624638	2.047625027
11677	4	2.591635801	2.602068714	11677	4	2.903830552	2.934334312

Table 7: Top 25 gene indices before and after estimation for GDS4382.

Rank	Ranking	
	Gene indices prior to missing value insertion	Gene indices after missing value insertion
1	714	714
2	1245	1245
3	1578	1578
4	1763	1763
5	2792	2792
6	4025	4025
7	4134	4134
8	5082	5082
9	8426	8426
10	9979	9979
11	10083	10083
12	10145	10145
13	10208	10208
14	10280	10280
15	10323	10323
16	10725	10725
17	10789	10789
18	10855	10855
19	11050	11050
20	11055	11055
21	11100	11100
22	11465	11465
23	11485	11485
24	11650	11650
25	11677	11677
Number of common genes in top 25 positions = 25		
% of common genes = 100		

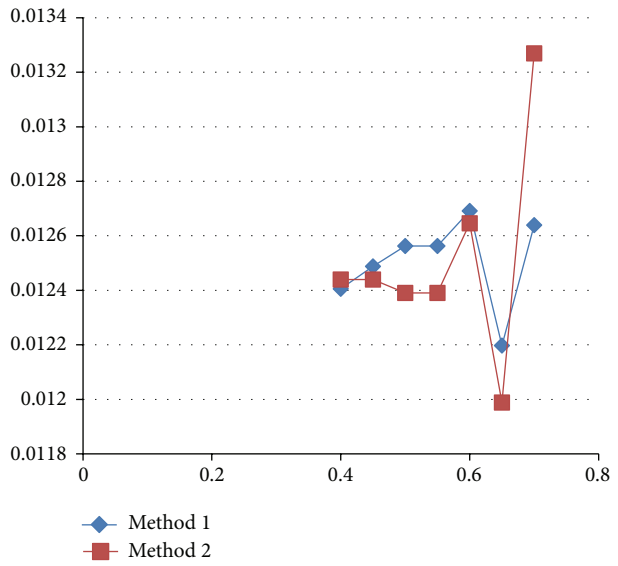

Figure 3: Plot of RMSE versus membership grade θ for alpha dataset.

Table 8: Estimated values for Breast Cancer dataset.

Breast Cancer							
Cancer				Normal			
Missing values inserted		Old_value	Estimated value with mem = 0.55	Missing values inserted		Old_value	Estimated value with mem = 0.55
At row	At column			At row	At column		
272	12	−0.354039943	−0.419709522	272	8	−0.407057103	−0.396601672
329	5	−0.176687651	−0.295980517	329	2	−0.021584122	0.092981006
491	3	0.222486126	0.30260015	491	10	0.140530363	0.067353238
869	11	0.79646566	0.852100043	869	5	0.345360946	0.279331888
1143	9	0.017956445	0.256958128	1143	4	0.054582833	−0.145618319
1937	1	0.22672464	0.33943153	1937	7	−0.128257731	0.05338273
2825	8	7.552441375	8.108907291	2825	3	6.080319682	5.969522121
3004	2	0.128475064	0.04404869	3004	6	−0.124353092	−0.113116385
4911	4	−0.550612392	−0.372472073	4911	9	−0.090276567	−0.152051028
5328	6	10.57228289	9.556257324	5328	5	8.316072021	7.887609502
6184	11	−0.493537621	−0.504575199	6184	8	0.035382884	−0.132574115
7941	10	0.233974208	0.14984931	7941	6	0.077199916	0.101298927
8452	14	−0.312716903	−0.32992237	8452	3	−0.071344506	−0.092371778
9076	6	0.708996621	0.154489296	9076	1	0.060554625	0.036078623
9267	13	−0.270092957	−0.33842598	9267	10	−0.034956721	0.30142636
9574	7	0.093118778	0.074772718	9574	7	−0.227882015	−0.159726559
9723	4	−0.018883445	−0.279787171	9723	5	0.407076237	0.815509208
9753	10	−0.23265185	−0.264909557	9753	8	−0.228475796	−0.265794544
9905	3	−0.31537376	−0.417539213	9905	3	−0.286431974	−0.199211435
10319	8	−0.049692673	−0.038561161	10319	9	−0.218073382	−0.242557019
10614	9	−0.511734814	−0.430792872	10614	2	−0.268390734	−0.1922339
11377	1	−0.055809992	0.083662727	11377	7	−0.268295765	−0.144527083
11737	12	−0.34275829	−0.269422135	11737	4	−0.387083296	−0.093221982
11976	6	0.030992978	−0.133189906	11976	9	−0.270598103	−0.272899078
12053	4	−0.374640827	−0.328380483	12053	10	−0.360077886	−0.244255658

Table 9: Top 25 gene indices before and after estimation for Breast Cancer dataset.

Rank	Ranking	
	Gene indices prior to missing value insertion	Gene indices after missing value insertion
1	3004	1143
2	7941	3004
3	10319	7941
4	869	10319
5	5328	11737
6	9723	491
7	491	9753
8	9574	869
9	9905	5328
10	11737	9723
11	2825	12053
12	4911	2825
13	8452	9574
14	272	9905
15	329	4911
16	6184	8452
17	9076	9076
18	9267	329
19	9753	6184
20	10614	9267
21	11377	11976
22	11976	2218
23	12053	2459
24	1143	2995
25	1937	4200
Number of common genes in top 25 positions = 21		
% of common genes = 84		

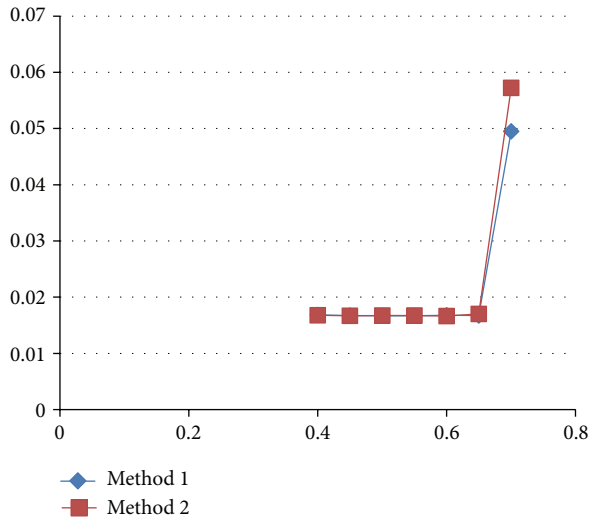

Figure 4: Plot of RMSE versus membership grade θ for cdc15 dataset.

Table 10: Estimated values for Prostate Cancer dataset.

Prostate Cancer							
Cancer				Normal			
Missing values inserted		Old_value	Estimated value with mem = 0.55	Missing values inserted		Old_value	Estimated value with mem = 0.55
At row	At column			At row	At column		
205	45	7.890085103	9.434976702	205	45	5.168380524	5.416011322
2839	5	−0.176203136	0.004447962	2839	5	−0.173585261	−0.183852307
3649	10	0.296457522	0.046654036	3649	10	0.329958724	0.617420849
3794	8	−0.241993865	−0.216817937	3794	8	−0.106322335	−0.109294537
4365	17	0.059874409	2.443242853	4365	17	−0.122967971	−0.099879376
5757	14	0.090277183	0.70723886	5757	14	−0.196702226	−0.067191045
5944	22	−0.180239863	−0.186977867	5944	22	−0.137411352	−0.217659599
6185	36	1.557345019	1.548681099	6185	36	−0.010802226	−0.181261903
6462	28	−0.16997265	−0.175762613	6462	28	0.202288499	0.191383577
7247	32	0.859643204	0.956551639	7247	32	0.932414697	2.21258573
7520	18	0.270631813	0.179429504	7520	18	0.064953651	0.155169008
7557	11	0.393712194	0.375941042	7557	11	1.450975133	1.084180805
7768	3	0.042090452	0.11563321	7768	3	0.59875936	0.576608917
8123	47	0.216968028	0.283330867	8123	47	0.128003038	0.165883246
8554	4	0.015161421	0.083268762	8554	4	0.40771258	−0.051156112
8768	26	−0.135452978	−0.04501778	8768	26	−0.138712043	−0.188517852
8850	17	2.708006948	1.916254815	8850	17	−0.364936796	−0.228740529
9034	29	−0.190217605	−0.178815842	9034	29	−0.194249889	−0.047431811
9050	34	0.351634344	0.174924563	9050	34	0.284389403	0.469260339
9172	42	2.277059356	2.347704101	9172	42	1.727648712	1.562259352
9850	50	−0.046243874	−0.290856086	9850	50	1.371476261	2.134924482
10138	14	0.599428228	0.628396963	10138	14	0.765017083	1.001711406
10494	22	−0.189181177	−0.122575727	10494	22	0.220022109	0.365466432
10537	48	0.139373755	0.406906442	10537	48	0.035059438	0.011805144
10956	7	0.337716243	0.155279221	10956	7	0.146282156	0.169249764

Table 11: Top 25 gene indices before and after estimation for Prostate Cancer dataset.

Rank	Ranking	
	Gene indices prior to missing value insertion	Gene indices after missing value insertion
1	6185	6185
2	10494	10494
3	9850	4365
4	4365	9850
5	10138	9034
6	9172	10138
7	9034	5944
8	5944	9172
9	3649	3649
10	8554	2839
11	2839	7557
12	7557	10956
13	205	9050
14	3794	7520
15	10956	3794
16	8850	205
17	7520	8850
18	9050	10537
19	10537	5757
20	5757	8554
21	8123	8768
22	6462	8123
23	8768	6462
24	7247	7247
25	7768	9093
Number of common genes in top 25 positions = 24		
% of common genes = 96		

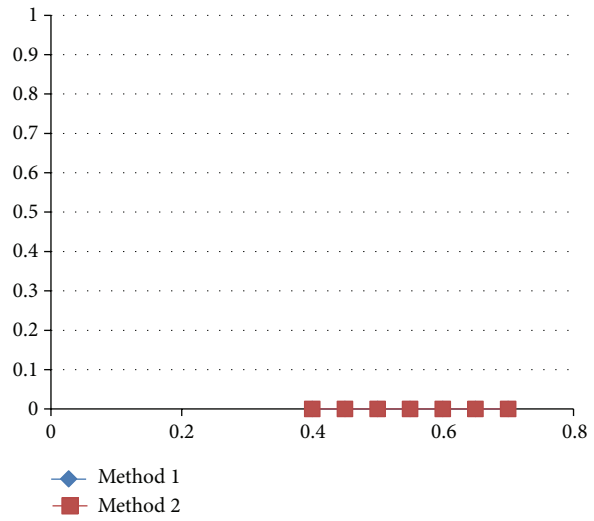

Figure 5: Plot of RMSE versus membership grade θ for elu dataset.

Table 12: Performance comparison for Prostate dataset.

Dataset name				Prostate			
Number of genes	Number of samples	Number of samples in Cancer dataset	Number of samples in normal dataset	Number of samples in training dataset	Number of samples in testing dataset	SVM kernel used	Number of folds for LOOCV
12600	102	52	50	82 (42 cancer, 40 normal)	20 (10 cancer, 10 normal)	Linear	41
Algorithm		% accuracy		Sensitivity/recall	Specificity	*F*1-score	*G*-mean
Proposed approach		100 (with top 5 genes)		1	1	1	1
PSO based graph theoretic approach		91		0.91	0.92	0.91	0.91
DWFS using KNN classifier		86		0.87	0.85	0.86	0.86
DWFS using NBC classifier		80		0.76	0.85	0.78	0.80

Table 13: Estimated values for DLBCL-FL.

Leukaemia (DLBCL-FL)							
Cancer				Normal			
Missing values inserted		Old_value	Estimated value with mem = 0.55	Missing values inserted		Old_value	Estimated value with mem = 0.55
At row	At column			At row	At column		
28	50	−0.188603483	−0.203477278	28	2	3.886811534	0.84848513
447	45	0.103798973	0.304241026	447	18	−0.146274929	0.009235547
546	8	1.055798216	1.246420135	546	8	0.140697423	0.320415944
640	52	0.052220359	−0.311252744	640	7	0.12552933	0.364757746
913	5	0.15759485	0.180411915	913	2	−0.135855848	−0.165390479
1129	4	1.051658222	0.598949762	1129	10	0.04903928	0.096097027
1142	31	−0.249901255	−0.311776993	1142	1	−0.249625598	−0.196558516
1293	34	1.91037315	2.330011528	1293	3	0.27829269	0.224522592
1553	48	−0.218067263	−0.107003318	1553	9	−0.00577268	0.602953445
1731	26	0.555312243	0.39207635	1731	7	0.448099319	0.524633075
2062	22	2.366047008	4.45272374	2062	13	0.201613611	0.849245039
2929	17	0.072886107	0.125837311	2929	11	−0.007410482	0.004016071
3965	56	0.137340738	−0.905551574	3965	16	−0.133906259	−0.083702381
3969	36	2.328255836	2.60971631	3969	6	0.520731724	0.584352541
4124	17	0.077170048	0.183068221	4124	5	−0.157303536	−0.036843328
4135	10	−0.35554626	−0.339641516	4135	10	−0.232750407	−0.161598404
4143	14	0.185986189	−0.024609518	4143	7	−0.032712551	0.024485269
4233	18	3.990305253	1.582890447	4233	8	0.770838584	0.706726475
4313	11	0.453186971	0.126025314	4313	15	−0.129336617	−0.079142717
4510	22	−0.167456263	−0.135667934	4510	17	−0.046588554	−0.100490086
5327	46	−0.415159893	−0.324486992	5327	12	−0.146934163	−0.22684469
6120	3	4.432535499	4.497409395	6120	3	0.919305283	0.723423487
6417	7	−0.253214481	−0.312493286	6417	10	−0.141004928	−0.007636336
6434	42	−0.220525385	−0.208472395	6434	14	−0.001659417	0.063668162
6756	55	2.0189449	1.307321937	6756	4	−0.009878439	0.570901233

Table 14: Top 25 gene indices before and after estimation for DLBCL-FL.

Rank	Ranking	
	Gene indices prior to missing value insertion	Gene indices after missing value insertion
1	447	447
2	913	4135
3	4135	913
4	546	640
5	2929	2929
6	640	1142
7	4510	546
8	3969	3969
9	5327	4510
10	6756	4233
11	4233	5327
12	4313	4313
13	6120	6756
14	1142	6120
15	1129	1553
16	1731	1129
17	4124	6417
18	3965	4124
19	1293	1731
20	6434	1293
21	28	6434
22	4143	28
23	2062	2062
24	1553	4094
25	6417	1984
Number of common genes in top 25 positions = 23		
% of common genes = 92		

5. Conclusion and Future Scope

The proposed modified version of LRFDVImpute technique has been tested on the dataset from Spellman et al. [26] and has shown impressive results. It outperforms some state-of-the-art methods. The plots of RMSE versus membership grade θ show that modified version is equivalent to or better than earlier version for the alpha and cdc15 datasets. However, for the cdc28 dataset, earlier version has shown better results. For the elu datasets, both have reached 0 error margin. For both versions, a membership grade between 0.55 and 0.65 produces minimum error and any value in this range can be considered as a threshold to be used for fresh experiments.

The validation of the missing value estimation shows that most of the top ranked genes remain the same, before and after imputation, which implies that the proposed modified LRFDVImpute technique has been accurate in estimating the unknown values.

As a future scope, we would like to analyze the effects of using quadratic regression for estimation of missing values and the use of data cleaning techniques before imputation which may remove outliers if any and may further reduce the error margin. For gene ranking, we wish to analyze the effects of different parameter settings for GA and observe the ranking and classification results using SVM with other kernels and also compare results with the ones mentioned in literature. We would also wish to modify our algorithms so as to make this ranking more efficient and find out the most significant genes that would correctly identify the subtypes of a particular type of cancer. For the Leukemia dataset [16], this could be identifying the B-cell and T-cell lineages for the acute lymphoblastic leukemia (ALL) samples.

Competing Interests

The authors declare that they have no competing interests.

Table 15: Performance comparison for DLBCL-FL.

Dataset name				DLBCL-FL			
Number of genes	Number of samples	Number of samples in Cancer dataset	Number of samples in normal dataset	Number of samples in training dataset	Number of samples in testing dataset	SVM kernel used	Number of folds for LOOCV
7070	77	58	19	62 (47 DLBCL, 15 FL)	15 (11 DLBCL, 4 FL)	Linear	31
Algorithm		% accuracy		Sensitivity/recall	Specificity	*F*1-score	*G*-mean
Proposed approach		100 (with top 5, 10, 15, and 20 genes)		1	1	1	1
PSO based graph theoretic approach		94		0.95	0.94	0.89	0.94
DWFS using KNN classifier		91		0.97	0.85	0.94	0.9
DWFS using NBC classifier		96		1	0.9	0.98	0.94

References

[1] O. Troyanskaya, M. Cantor, G. Sherlock et al., "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.

[2] S. Saha, P. K. Singh, and K. N. Dey, "Missing value estimation in DNA microarrays using linear regression and fuzzy approach," in *Proceedings of the 4th International Conference on Advances in Computer Science and Application (CSA '15)*, pp. 62–70, World Scientific, Thiruvananthapuram, India, October 2015.

[3] S. Saha, K. N. Dey, R. Dasgupta, A. Ghose, and K. Mullick, "Anirban ghose, and koustav mullick: missing value estimation in DNA microarrays using B-splines," *Journal of Medical and Bioengineering*, vol. 2, no. 2, pp. 88–92, 2013.

[4] L. C. Crossman, M. Mori, Y.-C. Hsieh et al., "In chronic myeloid leukemia white cells from cytogenetic responders and non-responders to imatinib have very similar gene expression signatures," *Haematologica*, vol. 90, no. 4, pp. 459–464, 2005.

[5] Graham Hole Research Skills, *The Wilcoxon Test*, Version 1.0, 2011.

[6] O. Alter, P. O. Brown, and D. Botstein, "Singular value decomposition for genome-wide expression data processing and modeling," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 18, pp. 10101–10106, 2000.

[7] G. H. Golub and C. F. V. Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.

[8] H. Kim, G. H. Golub, and H. Park, "Missing value estimation for DNA microarray gene expression data: local least squares imputation," *Bioinformatics*, vol. 21, no. 2, pp. 187–198, 2005.

[9] S. Oba, M.-A. Sato, I. Takemasa, M. Monden, K.-I. Matsubara, and S. Ishii, "A Bayesian missing value estimation method for gene expression profile data," *Bioinformatics*, vol. 19, no. 16, pp. 2088–2096, 2003.

[10] Z. Bar-Joseph, G. K. Gerber, D. K. Gifford, T. S. Jaakkola, and I. Simon, "Continuous representations of time-series gene expression data," *Journal of Computational Biology*, vol. 10, no. 3-4, pp. 341–356, 2003.

[11] S. Chakraborty, S. Saha, and K. Dey, "Missing value estimation in DNA microarray—a fuzzy approach," *International Journal of Artificial Intelligence and Neural Networks (IJAINN)*, vol. 2, no. 1, 2012.

[12] C. Yooa, I. B. Leeb, and P. A. Vanrolleghema, "Interpreting patterns and analysis of acute leukemia gene expression data by multivariate fuzzy statistical analysis," *Computers & Chemical Engineering*, vol. 29, no. 6, pp. 1345–1356, 2005.

[13] L. E. Peterson and M. A. Coleman, "Comparison of gene identification based on artificial neural network pre-processing with k-means cluster and principal component analysis," in *Fuzzy Logic and Applications*, I. Bloch, A. Petrosino, and A. G. B. Tettamanzi, Eds., vol. 3849 of *Lecture Notes in Computer Science*, pp. 267–276, 2006.

[14] C. Liao, S. Li, and Z. Luo, "Gene selection using Wilcoxon rank sum test and support vector machine for cancer classification," in *Computational Intelligence and Security*, Y. Wang, Y.-M. Cheung, and H. Liu, Eds., vol. 4456 of *Lecture Notes in Computer Science*, pp. 57–66, 2007.

[15] M. West, C. Blanchette, H. Dressman et al., "Predicting the clinical status of human breast cancer by using gene expression profiles," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 20, pp. 11462–11467, 2001.

[16] T. R. Golub, D. K. Slonim, P. Tamayo et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–527, 1999.

[17] E. B. Huerta, B. Duval, and J.-K. Hao, "A hybrid GA/SVM approach for gene selection and classification of microarray data," in *Applications of Evolutionary Computing*, F. Rothlauf, J. Branke, S. Cagnoni et al., Eds., vol. 3907 of *Lecture Notes in Computer Science*, pp. 34–44, Springer, Berlin, Germany, 2006.

[18] U. Alon, N. Barka, D. A. Notterman et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, no. 12, pp. 6745–6750, 1999.

[19] K. C. Mondal, A. Mukhopadhyay, U. Maulik, S. Bandhyapadhyay, and N. Pasquier, "MOSCFRA: a multi-objective genetic approach for simultaneous clustering and gene ranking," in *Computational Intelligence Methods for Bioinformatics and Biostatistics*, R. Rizzo and P. J. G. Lisboa, Eds., vol. 6685 of *Lecture Notes in Computer Science*, pp. 174–187, Springer, Berlin, Germany, 2011.

[20] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[21] R. M. Luque-Baena, D. Urda, J. L. Subirats, L. Franco, and J. M. Jerez, "Analysis of cancer microarray data using constructive neural networks and genetic algorithms," in *Proceedings of the 1st International Work-Conference on Bioinformatics and Biomedical Engineering-IWBBIO*, Granada, Spain, March 2013.

[22] M. Mandal and A. Mukhopadhyay, "A novel PSO-based graph-theoretic approach for identifying most relevant and non-redundant gene markers from gene expression data," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 30, no. 3, pp. 175–192, 2015.

[23] O. Soufan, D. Kleftogiannis, P. Kalnis, and V. B. Bajic, "DWFS: a wrapper feature selection tool based on a parallel genetic algorithm," *PLoS ONE*, vol. 10, no. 2, Article ID e0117988, 2015.

[24] J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, UK, 2nd edition, 1970.

[25] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, New York, NY, USA, 3rd edition, 1996.

[26] P. T. Spellman, G. Sherlock, M. Q. Zhang et al., "Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization," *Molecular Biology of the Cell*, vol. 9, no. 12, pp. 3273–3297, 1998.

[27] A. Khamas, T. Ishikawa, K. Shimokawa et al., "Screening for epigenetically masked genes in colorectal cancer using 5-aza-2′-deoxycytidine, microarray and gene expression profile," *Cancer Genomics and Proteomics*, vol. 9, no. 2, pp. 67–75, 2012.

[28] T. Sato, A. Kaneda, S. Tsuji et al., "PRC2 overexpression and PRC2-target gene repression relating to poorer prognosis in small cell lung cancer," *Scientific Reports*, vol. 3, article 1911, 2013.

[29] D. Singh, P. G. Febbo, K. Ross et al., "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell*, vol. 1, no. 2, pp. 203–209, 2002.

[30] M. A. Shipp, K. N. Ross, P. Tamayo et al., "Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning," *Nature Medicine*, vol. 8, no. 1, pp. 68–74, 2002.

[31] M. H. Cheok, W. Yang, C.-H. Pui et al., "Treatment-specific changes in gene expression discriminate in vivo drug response

in human leukemia cells," *Nature Genetics*, vol. 34, no. 1, pp. 85–90, 2003.

[32] J. C. Chang, E. C. Wooten, A. Tsimelzon et al., "Gene expression profiling for the prediction of therapeutic response to docetaxel in patients with breast cancer," *The Lancet*, vol. 362, no. 9381, pp. 362–369, 2003.