Approach the Interval Type-2 Fuzzy System and PSO Technique in Landcover Classification

Dinh Sinh Mai(✉), Long Thanh Ngo, and Le Hung Trinh

Le Quy Don Technical University, Hanoi, Vietnam
maidinhsinh@gmail.com

Abstract. In fuzzy classification systems, the estimation of the optimal number of clusters and building base-rules are very important and greatly affects the accuracy of the fuzzy system. Base-rules are often built on the experience of experts, but this is not always good and the results are often unstable. Particle swarm optimization (PSO) techniques have many advantages in finding optimal solutions and have been used successfully in many practical problems. This paper proposes a method using the PSO technique to build base-rules for the interval type-2 fuzzy system (IT2FS). Experiments performed on satellite image data for the landcover classification problem have shown that the proposed method works more stably and effectively than the non-PSO technique.

Keywords: Type-2 fuzzy set · Interval type-2 fuzzy system · PSO · Fuzzy system · Landcover · Satellite image

1 Introduction

Type-2 fuzzy set (T2F set) is an extension from the type-1 fuzzy set (T1F set), which is characterized by the fuzzy membership function [1], unlike T1F set, membership function (MF) values are a clear number of [0, 1], MF values of the T2F set are a fuzzy set of [0, 1] [2]. One of the applications of the T2F set is the interval type-2 fuzzy c-means (IT2FCM) clustering algorithm [3], which has been used in many practical problems [4, 5]. However, this is an unsupervised clustering algorithm and they have difficulty in automatic classification [6].

T2FSs are characterized by a three-dimensional fuzzy MF including the footprint of uncertainty (FOU) that can directly model and handle the uncertainty [7]. Once the type-1 MF is selected, all uncertainties will disappear because the type-1 MFs are completely correct [8]. The type-2 fuzzy system (T2FS) based on the T2F set has been used in many practical applications such as predictive problem [9], industrial control [10, 11], data classification [12, 13].

Satellite images with many advantages such as wide coverage, fast update times have been applied in many fields [20]. However, satellite image data also has many uncertainties that use clustering algorithms that are often ineffective [21, 22]. Although T2FS has been widely applied in many areas [14], according to the author's knowledge, applications in problems related to remote sensing data is still very few and mainly based

N. T. Nguyen et al. (Eds.): ACIIDS 2020, LNAI 12034, pp. 402–414, 2020.
https://doi.org/10.1007/978-3-030-42058-1_34

on T1F set, so T2FS-based approach to the problem of remote sensing image processing is a potential research direction. Moreover, due to the complexity of the calculation, it limits T2FS in real applications. One of the case of T2FS is more widely used, which is an interval type-2 fuzzy system (IT2FS) [15]. This study introduces an approach of IT2FS in the remote sensing image landcover classification.

Currently, there are many optimization methods that do not need to use the derivative of objective functions. However, the disadvantage of using the derivative besides complex calculations of derivative formulas, when the UMF or LMF changes the mathematical formula on the specified domain, the calculation will also change; moreover, they are easily stuck at a local extreme [16, 17].

Methods that do not use derivatives are often called evolutionary methods [8] or methods of biological inspiration [19] such as evolutionary programming, genetic algorithms, genetic programming, particle swarm optimization (PSO), quantum particle swarm optimization (QPSO), simulated annealing, differential evolution, ant colony optimization, gravitational search, so on.

These methods tend to be stronger than derivative-based methods because the process of finding a globally optimal solution is repeated many times until convergence. This can be used to optimize the FOU parameter in the IT2 fuzzy system [16]. With so many biological-inspired methods, which method is good to optimize the parameters of the fuzzy system? Each algorithm has advantages and disadvantages, so far no algorithm has been proven to be the best. Which algorithm is used depends on each specific problem and user familiarity [19].

However, these bio-inspired methods often have to use a large number of loops to find the optimal solution, they need a large amount of time to evaluate the objective function for each candidate. If the calculation time is not important, these are very powerful methods. The advantage of PSO algorithm is convergence faster than GA algorithm, which is suitable for large data sets such as satellite image data.

The fuzzy systems generally have difficulty in determining MF values, FOU, the number of rules [8]. When the MF values, the number of inputs and outputs are selected, the IT2FS can be built. Determining these parameters for fuzzy systems is very important and greatly affects the accuracy of the fuzzy system. In the paper, we use the PSO techniques to find the MF parameters for IT2FS.

The paper includes five sections following: Sect. 1 Introduction; Sect. 2 Background; Sect. 3 Method proposed; Sect. 4 Experimental and Sect. 5 Conclusion.

2 Backgrounds

2.1 Interval Type-2 Fuzzy Logic System

The IT2FS is characterized by interval type-2 fuzzy sets (IT2F sets) [8] consisting of 5 main parts as shown in Fig. 1.

There are two types of IT2FSs as shown in Fig. 1 (a and b), one of which is type-reduction and then defuzzification, the second is direct defuzzification. However, in practice the type combines both type-reduction and defuzzification to be more widely used because of lower computational complexity.

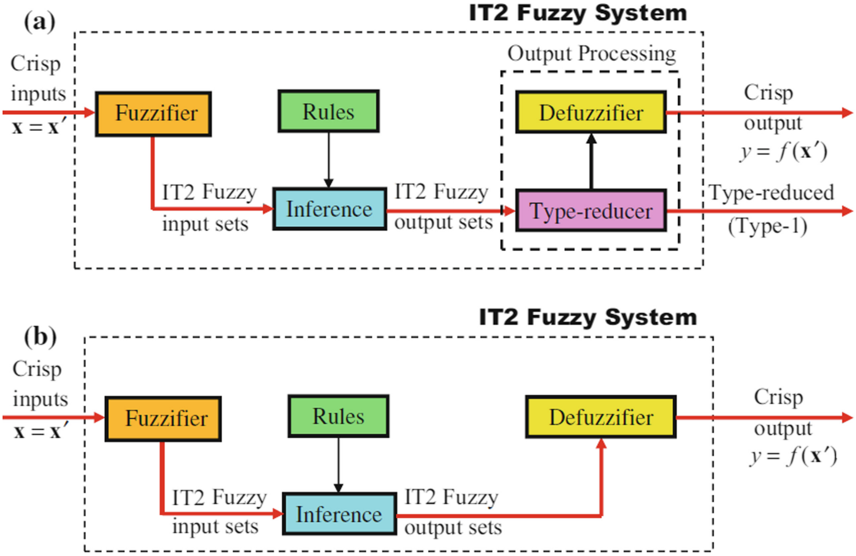

Fig. 1. Interval type-2 fuzzy system architecture [8]

The IT2FS works as follows, the crisp inputs are the attributes of the initial data, which fuzzifier into the input IT2F sets and then activate the inference engine and rule base to maps input IT2F sets into output IT2F sets. These output IT2F sets are then processed by the type-reducer to obtain T1F sets (type reducers). The defuzzifier then defuzzify output T1F sets to create the crisp output.

+ *Fuzzifier*

With T1FS, two types of fuzzifiers are used as singleton and non-singleton, meanwhile with T2FS, there are 3 types of fuzzifiers used including singleton, type-1 non-singleton, and IT2 non-singleton [11]. The fuzzifier maps a crisp input will depend on the choice of the type of fuzzifier. Assuming there are n inputs $X = (x_1, x_2, \ldots, x_n) \in X_1 x X_2 x \ \ldots x X_n$, $\tilde{A}_x$ is a set of type-2 fuzzy inputs. For example, if $\tilde{A}_x$ is a type-2 fuzzy singleton fuzzifier, then $\mu_{\tilde{A}(x_i)} = 1/1$ when $x_i = x_i'$ and $\mu_{\tilde{A}(x_i)} = 1/0$ when $x_i \neq x_i'$ and $x_i \in X_i$

+ *Rule Base*

Consider the input $x_1 \in X_1, x_2 \in X_2, \ldots, x_n \in X_n$ and c output $y_1 \in Y_1, y_2 \in Y_2, \ldots, y_c \in Y_c$. The rules of T2FS are similar to those of T1FS, only different the antecedents and the consequents instead of T1FS will be replaced by T2FS [11]:

$$R^i : \text{IF } x_1 \text{ is } \tilde{F}_1^i \text{ and } \ldots \text{and } x_n \text{ is } \tilde{F}_n^i \text{ THEN } y_1 \text{ is } \tilde{G}_1^i, \ \ldots, y_c \text{ is } \tilde{G}_c^i, \quad i = 1, \ \ldots, M \tag{1}$$

with M is the number of rules in the rule base.

+ *Fuzzy Inference Engine*

The inference engine and the rules that allow the mapping from input T2FS to the output T2FS. Each rule in a fuzzy rule base with M rules having n inputs $x_1 \in X_1, x_2 \in$

$X_2, \ldots, x_n \in X_n$ and output $y_k \in Y_k$, they can be written as follows:

$$R_k^i : \tilde{F}_1^i x \tilde{F}_2^i x \ldots x \tilde{F}_n^i \rightarrow \tilde{G}_k^i = \tilde{A}^i \rightarrow \tilde{G}_k^i \tag{2}$$

Where $\tilde{F}_j^i$ is the j^{th} T2FS, $j = 1, \ldots, n$, which is defined by a lower and upper bound membership function

$$\mu_{\tilde{F}_j^i}(x_j) = [\underline{\mu}_{\tilde{F}_j^i}(x_j), \bar{\mu}_{\tilde{F}_j^i}(x_j)], i = 1, \ldots, M; k = 1, \ldots, c \tag{3}$$

Compute the firing interval of the i^{th} rule, where * denotes the product operation.

$$\underline{f}^i(x) = \underline{\mu}_{\tilde{F}_1^i}(x_1) * \underline{\mu}_{\tilde{F}_2^i}(x_2) * \ldots * \underline{\mu}_{\tilde{F}_n^i}(x_n) \tag{4}$$

$$\bar{f}^i(x) = \bar{\mu}_{\tilde{F}_1^i}(x_1) * \bar{\mu}_{\tilde{F}_2^i}(x_2) * \ldots * \bar{\mu}_{\tilde{F}_n^i}(x_n) \tag{5}$$

+ *Type Reduction*

There are several algorithms used in type reduction, như Karnik-Mendel algorithm (KM) [15, 16], Enhanced Karnik-Mendel algorithm (EKM) [17], iterative algorithm and stopping condition (IASC), enhanced IASC algorithm (EIASC) [11]. In this study, the EIASC algorithm is used because they are easy to setup and the computational complexity is smaller than the remaining algorithms. Compute the output interval of the k^{th} fuzzy rule for the output, which is an interval T1FS $y_k = [y_{kl}, y_{kl}]$, the steps for calculating left most output y_{kl} and right most output y_{kr} using the EIASC algorithm are detailed below.

EIASC algorithm for calculating y_{kl} as follows:

Step 1: Sort $\underline{y}^i$ $(i = 1, \ldots, M)$ by increasing value $\underline{y}^1 \leq \underline{y}^2 \leq \ldots \leq \underline{y}^M$, note is $\underline{f}^i$ will also change the order corresponding to $\underline{y}^i$.
Step 2: Initialization

$$L = 0; a = \sum_{i=1}^{M} \underline{f}^i \underline{y}^i; b = \sum_{i=1}^{M} \underline{f}^i; \underline{y} = a/b \tag{6}$$

Step 3: Calculate:

$$\begin{aligned} &L = L + 1; a = a + \underline{y}^L(\bar{f}^{(L)} - \underline{f}^{(L)}) \\ &b = b + (\bar{f}^{(L)} - \underline{f}^{(L)}); \underline{y} = a/b \end{aligned} \tag{7}$$

Step 4: Stop condition: If $\underline{y} \leq \underline{y}^{(L+1)}$ then stop, otherwise go to Step 3.

Compute the k^{th} left most output:

$$y_{kl} = \frac{\sum_{u=1}^{L} \bar{f}^u \underline{y}^u + \sum_{v=L+1}^{M} \underline{f}^v \underline{y}^u}{\sum_{u=1}^{L} \bar{f}^u + \sum_{v=L+1}^{M} \underline{f}^v} \tag{8}$$

EIASC algorithm for calculating y_{kr} as follows:

Step 1: Sort $\bar{y}^i$ $(i = 1, \ldots, M)$ by increasing value $\bar{y}^1 \leq \bar{y}^2 \leq \ldots \leq \bar{y}^M$, note is $\bar{f}^i$ will also change the order corresponding to $\bar{y}^i$.
Step 2: Initialization

$$R = n; a = \sum_{i=1}^{M} \bar{f}^i \bar{y}^i; b = \sum_{i=1}^{M} \bar{f}^i; \bar{y} = a/b \tag{9}$$

Step 3: Calculate:

$$\begin{aligned} &a = a + \bar{y}^{(R)}(\bar{f}^{(R)} - \underline{f}^{(R)}); b = b + (\bar{f}^{(R)} - \underline{f}^{(R)}) \\ &\bar{y} = a/b;\ R = R - 1 \end{aligned} \tag{10}$$

Step 4: Stop condition: If $\bar{y} \geq \bar{y}^{(R+1)}$ then stop, otherwise go to Step 3.
Compute the k^{th} right most output:

$$y_{kr} = \frac{\sum_{u=1}^{R} \underline{f}^u \bar{y}^u + \sum_{v=R+1}^{M} \bar{f}^v \bar{y}^u}{\sum_{u=1}^{R} \underline{f}^u + \sum_{v=R+1}^{M} \bar{f}^v} \tag{11}$$

2.5 Defuzzification

The final crisp value of output of the IT2FS model is calculated by combining the corresponding outputs of M rules. For defuzzification solution we calculate the average left most point and right most point, therefore the crisp output for each output is calculated as follows:

$$Y_k(x) = \frac{y_{kl} + y_{kr}}{2} (k = 1, \ldots, c) \tag{12}$$

2.2 Particle Swarm Optimization

The PSO is an adaptive evolution algorithm based on finding the optimal solution for the population, the idea of algorithms comes from the hunting behavior of the birds [19]. Each problem will converge at one or several optimal solutions in the search space, considering each individual is a particle and a set of particles will be a population.

Each state of the population in the search space is considered as a candidate solution, the optimal solution is found by moving particles in the search space according to the position and velocity of the particle as the following formula:

$$\begin{aligned} &vt_i^{k+1} = \omega * vt_i^k + c_1 * r_1 * (P_{ibest} - v_i^k) + c_2 * r_2 * (G_{ibest} - v_i^k) \\ &v_i^{k+1} = v_i^k + vt_i^{k+1} \end{aligned} \tag{13}$$

In which, v_i^k is position of individual i^{th} in k^{th} generation, vt_i^k is velocity of individual i^{th} in k^{th} generation, ω is coefficient of inertia, c_1, c_2 is the acceleration coefficient, with a value of 1.5 to 2.5; r_1, r_2 is the random number, with values in the range [0,1].

In each loop, the optimal position search is performed by updating the velocity and position of the individual. In addition to each loop, the target value of each individual location is determined by an objective function.

3 Proposal Method

The paper developed a method using PSO technique to optimize MFs parameters of IT2FS. Some membership functions are often used in fuzzy systems such as triangular, trapezoidal, Gaussian, Cauchy, Laplace. In this study, we use Gaussian functions to build the MFs for IT2FS (see Fig. 2).

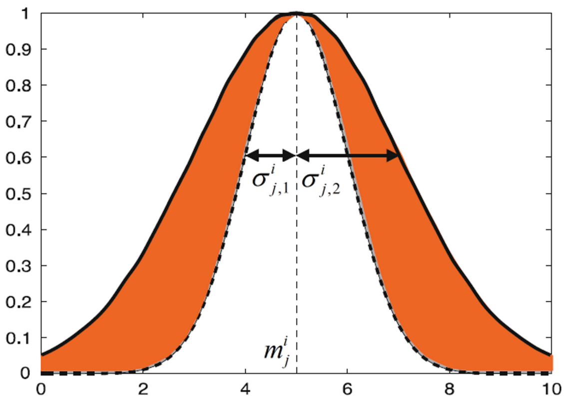

Fig. 2. Example of Gaussian IT2 MF

$$\mu_{\tilde{F}^i_j}(x_j) = \exp\left(-\frac{1}{2}\left(\frac{x_j - m^i_j}{\sigma^i_j}\right)^2\right) = \mathrm{N}(\mathrm{x}_j, m^i_j, \sigma^i_j) \text{ with } \sigma^i_j \in \left[\sigma^i_{j,1}, \sigma^i_{j,2}\right] \quad (14)$$

It can be seen that the Gaussian function is characterized by parameters m^i_j, $\sigma^i_{j,1}$, $\sigma^i_{j,2}$. The number of Gaussian functions used will be equal to the number of landcovers. If c is the number of overlays to be classified, then the number of parameters is 3 * c.

The idea of the paper is from labeled data sets, using PSO technique to find the optimal parameters of the Gaussian function for unknown fuzzy systems. Each of the above parameters can be considered as an individual in a population, each individual will include position and velocity moving in the search space. The PSO algorithm will stop if the position of the individual is optimal or when the number of loops is satisfied.

To evaluate the effectiveness of the proposed method, we measure the difference between the actual output and the desired output on the data sets labeled by the following formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2 \quad (15)$$

In PSO, individuals will move through the search space and record optimal locations. PSO consists of 3 main steps: population initialization; evaluate individuals; update position and velocity for individuals (use the formula (13)). Each particle will include position and velocity, which is limited to the search space $v_{\min} \leq v_i^k \leq v_{max}$ and $vt_{\min} \leq vt_i^k \leq vt_{max}$; in case if $v_i^k < v_{\min}(v_i^k > v_{\max})$ then $v_i^k = v_{\min}(v_i^k = v_{\max})$; if $vt_i^k < vt_{\min}(vt_i^k > vt_{max})$ then $vt_i^k = vt_{\min}(vt_i^k = vt_{max})$.

The optimal process is considered by the two parameters P_{best} and G_{best}, P_{best} is the best solution of the individual at the present location and G_{best} is the best solution of the population at the current location. These values will be updated based on the objective function value (15). In each iteration of the PSO algorithm, if a position of the particles optimizes the objective function (15) (the smaller value), the position of the particle is saved by P_{best}; If the position of the particles makes the objective function (15) reach the minimum value, the position of that particle will be saved by G_{best}. The parameters of the PSO algorithm including $c_1, c_2, r_1, r_2, \omega$ will be selected based on the suggestion of the original paper.

The position of particles instead of random initialization will be initialized using the fuzzy c-mean algorithm (FCM) [18]. This may give the initial position of the individuals closer to the optimal value. Accordingly, the initial value of m_j^i is the centroid of the clusters; and the initial value of $\sigma_{j,1}^i, \sigma_{j,2}^i$ is calculated based on the standard deviation of the training data.

The use of PSO in fuzzy systems is as follows:

Algorithm 1: PSO

Step 1: The parameters of each MFs act as individuals in the population.
Step 2: The movement of individuals leads to the optimal solution of individuals and the entire population (considered by the objective function (15)).
Step 3: Once the parameters have been adjusted and it will be used to evaluate the performance of the fuzzy system.

The proposed method includes the following steps:

Algorithm 2: PSO in IT2FLS
Input: $X = (x_1, x_2, \ldots, x_n)$, number of rules M, number of Gaussian functions.

Step 1: Initialize parameters for the Gaussian MFs: Perform the FCM algorithm to initialize m_j^i and using labeled data to initialize the standard deviation.
Step 2: Compute the lower and upper membership function of x_j:

$$\mu_{\tilde{F}_j^i}(x_j) = [\underline{\mu}_{\tilde{F}_j^i}(x_j), \bar{\mu}_{\tilde{F}_j^i}(x_j)]$$

Step 3: Compute the firing interval of the i^{th} rule, where * denotes the product operation.

$$\underline{f}^i(x) = \underline{\mu}_{\tilde{F}_1^i}(x_1) * \underline{\mu}_{\tilde{F}_2^i}(x_2) * \ldots * \underline{\mu}_{\tilde{F}_n^i}(x_n)$$

$$\bar{f}^i(x) = \bar{\mu}_{\tilde{F}_1^i}(x_1) * \bar{\mu}_{\tilde{F}_2^i}(x_2) * \ldots * \bar{\mu}_{\tilde{F}_n^i}(x_n)$$

Step 4: Compute the left most and right most output of the i^{th} fuzzy rule:

$$y^i = [y_l^i, y_r^i]$$

Step 5: The crisp output for each output is calculated as follows: $Y(x) = \frac{y_l + y_r}{2}$
Step 6: Compute MSE: $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2$
Step 7: If $(MSE < \varepsilon)$ or $(Loop > Loop_{max})$ goto step Output.
Step 8: Implementing Algorithm 1 and goto step 2.

Output: The parameters for IT2FLS.

The complexity of the proposed IT2FLS will also include the complexity of FCM in step 1, the complexity of PSO in step 8. After each implementation of IT2FLS, the parameters of the Gaussian function are adjusted by the PSO algorithm, the process is repeated until the optimal parameter for the fuzzy system is found. After the training is completed, the parameters of the MFs will be used for the fuzzy system.

Because PSO has smaller computational complexity than genetic algorithm, evolutionary computation or deep learning, this method is able to find the optimal parameters faster. However, their disadvantage is that finding the best parameter depends on the selection of the parameters of the PSO algorithm.

4 Experimental Results

Experiment on three data sets downloaded from the UCI Machine Learning Repository [23] including 'Urban land cover data set', 'Crowdsourced Mapping Data Set' and 'Forest type mapping Data Set'. Detailed information about the test data sets are shown in Table 1.

Table 1. Experimental data

Data	Urban landcover	Crowdsourced	Forest type
Attributes	148	29	27
Training	168	10545	198
Testing	507	300	325

The resulting classification performance of the classification is evaluated by determining True Positive Rate (TPR) and False Positive Rate (FPR) defined as follows:

$$TPR = \frac{TP}{TP + FN} \text{ and } FPR = \frac{FP}{TN + FP} \tag{16}$$

Where TP is the number of correctly classified data and FN is the number of incorrectly misclassified data, FP is the number of incorrectly classified data and TN is the number of correctly misclassified data. The better the algorithm is, the higher the TPR value is and the smaller the FTR value is encountered.

4.1 Experiment 1

The experimental data set is 'Urban land cover data set' with the number of samples used for training 168 and used for testing 507.

The landcover classification results are shown in Tables 2 and 3. Accordingly, Table 2 is the accuracy by true positive rate and false positive rate, IT2FS-PSO method gives the highest accuracy with 92.12%, followed by 91.21% for IT2FS-FCM. The lowest accuracy with 84.94% for SVM method. Meanwhile, the lowest false positive rate is 1.18% for IT2FS-PSO, followed by 1.65%, 1.66%, 1.76%, 2.07% and 2.89% for IT2FS-FCM, T1FS-PSO, T1FS-FCM, KNN and SVM.

Table 2. Accuracy by true positive rate and false positive rate

Algorithm	KNN	SVM	T1FS-FCM	T1FS-PSO	IT2FS-FCM	IT2FS-PSO
TPR (%)	85.81	84.94	87.68	89.18	91.21	**92.12**
FPR (%)	2.07	2.89	1.76	1.66	1.65	**1.18**

Table 3. Accuracy according to the landcovers

Class	KNN	SVM	T1FS-FCM	T1FS-PSO	IT2FS-FCM	IT2FS-PSO
Trees	83.41	85.13	86.31	90.34	92.32	**92.34**
Grass	85.72	87.32	88.94	88.01	89.52	**90.58**
Soil	89.14	81.26	84.87	92.32	**96.71**	93.94
Concrete	87.31	83.62	89.72	91.56	93.36	**95.78**
Asphalt	80.52	84.31	87.54	89.19	88.63	**91.49**
Buildings	84.94	88.73	90.09	87.48	91.92	**94.63**
Cars	82.63	**89.52**	86.98	85.83	88.21	86.28
Pools	88.51	80.31	85.71	87.67	90.42	**92.76**
Shadows	90.13	84.23	88.93	90.23	89.83	**91.28**

The overview of Table 3 shows that the IT2FS-PSO method gives the highest accuracy on most landcovers. The overview of Table 3 shows that the IT2FS-PSO method gives the highest accuracy on most coatings. Only the "soil" class has the highest accuracy of 96.71% for IT2FS-FCM and the "cars" class has the highest accuracy of 89.52% for SVM.

4.2 Experiment 2

Crowdsourced data from OpenStreetMap is used to classify satellite images into different land cover classes (impervious, farm, forest, grass, orchard, water) with the number of samples used for training 10545 and used for testing 300.

Table 4 shows the highest accuracy of 90.30% for IT2FS-PSO method, but the lowest false positive rate belongs to T1FS-PSO method with 1.21%. In this experiment, most methods yielded an accuracy of less than 90%. In particular, the lowest accuracy is 87.88% for the KNN method.

Table 4. Accuracy by true positive rate and false positive rate

Algorithm	KNN	SVM	T1FS-FCM	T1FS-PSO	IT2FS-FCM	IT2FS-PSO
TPR (%)	87.88	88.20	88.18	89.12	89.11	**90.30**
FPR (%)	2.12	2.08	1.78	**1.21**	1.34	1.27

Table 5 shows the accuracy by landcovers. The IT2FS-PSO method achieved the highest accuracy for the "impervious", "farm" and "water" classes with 90.84%, 91.87%, and 92.61%, respectively. Meanwhile, T1FS-PSO method has the highest accuracy in the "forest" class with 88.93%; T1FS-FCM method has the highest accuracy in the "orchard" class with 89.03%; KNN method has the highest accuracy in the "grass" class with 90.23%.

Table 5. Accuracy according to the landcovers

Class	KNN	SVM	T1FS-FCM	T1FS-PSO	IT2FS-FCM	IT2FS-PSO
Impervious	87.89	89.21	85.89	87.38	87.38	**90.84**
Farm	90.93	89.49	87.39	90.27	90.99	**91.87**
Forest	85.59	87.65	88.43	**88.93**	88.81	88.38
Grass	**90.23**	86.48	88.21	89.84	88.45	89.45
Orchard	83.41	86.23	**89.03**	87.78	87.92	88.67
Water	89.22	90.12	90.12	90.49	91.08	**92.61**

In this experiment, the T1FS-PSO method gave an accuracy of 89.12% higher than the IT2FS-FCM method with 89.11%. Next to the SVM, T1FS-FCM and KNN methods. The overview can see, the difference in accuracy between the methods is not large.

4.3 Experiment 3

The experimental data set is the 'Mapping Data Set forest type' with the number of samples used for training 198 and used for testing 325.

Table 6 shows that the IT2FS-PSO method has the highest accuracy with 92.98%, while the following classification rate is only 0.98%. While the T1FS-FCM, T1FS-PSO, IT2FS-FCM methods all have accuracy above 90%, the KNN and SVM methods only reach 87.79% and 86.62%.

Table 6. Accuracy by true positive rate and false positive rate

Algorithm	KNN	SVM	T1FS-FCM	T1FS-PSO	IT2FS-FCM	IT2FS-PSO
TPR (%)	87.79	86.62	90.20	91.03	91.72	**92.98**
FPR (%)	2.13	2.38	1.72	1.07	1.19	**0.98**

Table 7. Accuracy according to the landcovers

Class	KNN	SVM	T1FS-FCM	T1FS-PSO	IT2FS-FCM	IT2FS-PSO
'Sugi'	91.34	89.16	90.78	90.32	90.71	**91.92**
'Hinoki'	89.22	87.51	90.82	89.94	91.53	**93.61**
'Mixed deciduous'	83.65	86.34	88.79	93.57	95.28	**95.32**
'Other' non forest land	86.93	83.48	90.41	90.29	89.34	**91.08**

Similarly on Table 7, the IT2FS-PSO method also gives the highest accuracy on all landcovers. The highest accuracy is 95.32% for the 'Mixed deciduous' class.

From the above experiments, it can be seen that the proposed method has the highest accuracy on most landcover. If the T1FS-FCM and IT2FS-FCM methods use the FCM algorithm to initialize parameters for the fuzzy system, the T1FS-PSO and IT2FS-PSO methods use FCM as the initial initialization step before executing the PSO algorithm to find optimal parameters for fuzzy systems. By using the PSO technique to find parameters for fuzzy systems, the landcover classification results achieve higher accuracy.

With the selection of the parameters of fuzzy system based on experience, the selection of parameters based on optimal techniques can help the classification algorithm be more stable, and achieve higher accuracy in most cases.

5 Conclusion

The paper proposed a method using the PSO technique to find the optimal solution for Gaussian MF parameters of IT2FS in landcover classification. Experimental results on 3 data sets from the UCI library show that the proposed method gives significantly higher accuracy than T1FS-FCM, T1FS-PSO, IT2FS-FCM, KNN and SVM methods. Moreover, PSO technology has a low computational complexity which is suitable for large data problems such as satellite image data.

In the next time, we will continue to study other optimization techniques such as evolutionary computation, deep learning for fuzzy classification systems.

Acknowledgment. This research is funded by the Newton Fund, under the NAFOSTED - UK Academies collaboration programme. This work was supported by the Domestic Master/PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF).

References

1. Karnik, N., Mendel, J., Liang, Q.: Type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. **7**(6), 643–658 (1999)
2. Mendel, J., John, R.: Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. **10**(2), 117–127 (2002)
3. Hwang, C., Rhee, F.C.H.: Uncertain fuzzy clustering: interval type-2 fuzzy approach to C-means. IEEE Trans. Fuzzy Syst. **15**(1), 107–120 (2007)
4. Ngo, L.T., Mai, D.S., Nguyen, M.U.: GPU-based acceleration of interval type-2 fuzzy C-means clustering for satellite imagery land-cover classification. In: 12th IEEE International Conference on Intelligent Systems Design and Applications (ISDA), pp. 992–997 (2012)
5. Mai, S.D., Ngo, L.T.: Interval type-2 fuzzy C-means clustering with spatial information for land-cover classification. In: Nguyen, N.T., Trawiński, B., Kosala, R. (eds.) ACIIDS 2015. LNCS (LNAI), vol. 9011, pp. 387–397. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15702-3_38
6. Ngo, L.T., Mai, D.S., Pedrycz, W.: Semi-supervising interval type-2 fuzzy C-means clustering with spatial information for multi-spectral satellite image classification and change detection. Comput. Geosci. **83**, 1–15 (2015)
7. Mendel, J.M.: Type-2 fuzzy sets and systems: an overview. IEEE Comput. Intell. Mag. **2**(1), 20–29 (2007)
8. Mendel, J.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, 2nd edn. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-51370-6
9. Abbas, K., Saeid, N., Doug, C., Dipti, S.: Interval type-2 fuzzy logic systems for load forecasting: a comparative study. IEEE Trans. Power Syst. **27**(3), 1274–1282 (2012)
10. Das, K., Suresh, S., Sundararajan, N.: A fully tuned sequential interval type-2 fuzzy inference system for motor-imagery task classification. In: IEEE International Conference on Fuzzy Systems, pp. 751–758 (2016)
11. Nguyen, T., Saeid, N.: Modified AHP for gene selection and cancer classification using type-2 fuzzy logic. IEEE Trans. Fuzzy Syst. **24**(2), 273–287 (2016)
12. Mendel, J., Liu, X.: Simplified interval type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. **21**(6), 1056–1069 (2013)
13. Liang, Q., Mendel, J.: Interval type-2 fuzzy logic systems: theory and design. IEEE Trans. Fuzzy Syst. **8**(5), 535–550 (2000)
14. Mendel, J.M., John, R.I., Liu, F.: Interval type-2 fuzzy logic systems made simple. EEE Trans. Fuzzy Syst. **14**(6), 808–821 (2006)
15. Karnik, N.N., Mendel, J.M.: Centroid of a type-2 fuzzy set. Inf. Sci. **132**(1–4), 195–220 (2001)
16. Karnik, N.N., Mendel, J.M.: Operations on type-2 fuzzy sets. Fuzzy Sets Syst. **122**(2), 327–348 (2001)
17. Wu, D., Mendel, J.M.: Enhanced Karnik-Mendel algorithms. IEEE Trans. Fuzzy Syst. **17**(4), 923–934 (2009)
18. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: the fuzzy C-means clustering algorithm. Comput. Geosci. **10**(2), 191–203 (1984)
19. Olivas, F., Angulo, L.A., Perez, J., Caraveo, C., Valdez, F., Castillo, O.: Comparative study of type-2 fuzzy particle swarm, Bee Colony and Bat algorithms in optimization of fuzzy controllers. Algorithms **10**, 1–27 (2017)
20. Mai, S.D., Ngo, L.T., Le Trinh, H.: Satellite image classification based spatial-spectral fuzzy clustering algorithm. In: Nguyen, N.T., Hoang, D.H., Hong, T.-P., Pham, H., Trawiński, B. (eds.) ACIIDS 2018. LNCS (LNAI), vol. 10752, pp. 505–518. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75420-8_48

21. Mai, D.S., Ngo, L.T.: General semi-supervised possibilistic fuzzy c-means clustering for land-cover classification. In: International Conference on Knowledge and Systems Engineering, pp. 1–6 (2019)
22. Dang, T.H., Mai, D.S., Ngo, L.T.: Multiple kernel collaborative fuzzy clustering algorithm with weighted super-pixels for satellite image land-cover classification. Eng. Appl. Artif. Intell. **85**, 85–98 (2019)
23. https://archive.ics.uci.edu/