Metaheuristic Tuning of Type-II Fuzzy Inference System for Data Mining

V. K. Ojha[1], Ajith Abraham[2], and V. Snášel[1]

[1]IT4Innovations, VŠB-Technical University of Ostrava Czech Republic
[2]Machine Intelligence Research Lab, Auburn, WA, USA

26 July 2016

Main Goal

1. To create diverse rules in Interval Type-2 Fuzzy Inference System
2. To reduce number of fuzzy rules in IT2FIS
3. To determine appropriate shape of type-2 fuzz sets
4. To analyze performance of the proposed IT2FIS optimization methods

Proposed optimization framework

1. Genetic algorithm based tuning of the knowledge-base of fuzzy inference system.

2. Swarm intelligence based tuning of the parameters of the rules in the knowledge-base.

Promising results: Comparisons of different models for Fridman dataset

Training set			Test set		
Rank	Models	RMSE	Rank	Models	RMSE
1	IT2FNN-SVR(N)	1.409	1	**IT2FIS-DE**	1.476
2	**IT2FIS-DE**	1.459	2	IT2FNN-SVR(F)	1.597
3	IT2FNN-SVR(F)	1.557	3	**IT2FIS-PSO**	1.766
4	**IT2FIS-PSO**	1.675	4	IT2FNN-SVR(N)	1.788
5	SEIT2FNN	1.841	5	SEIT2FNN	1.941
6	**IT2FIS-BFO**	1.948	6	**IT2FIS-BFO**	2.002
7	**IT2FIS-ABC**	2.053	7	**IT2FIS-ABC**	2.092
8	SONFIN	2.475	8	NBAG	2.121
9	T2FLS-G	2.534	9	GRNNFA	2.136
10	**IT2FIS-GWO**	2.667	10	Simple	2.224
11	NBAG	-	11	Bench	2.317
12	GRNNFA	-	12	SONFIN	2.531
13	Simple	-	13	T2FLS-G	2.597
14	Bench	-	14	**IT2FIS-GWO**	2.703

Improved results: For large learning iterations (10,000)

Table: Interval type-2 FIS training results over 80% data

Dataset	ABC	BFO	DE	GWO	PSO	IT2FNN-SVR(N)
Fridman	1.444	1.742	**1.360**	2.200	1.667	1.409

Models used for comparisons

Table: Models from literature

Model	Ref.	Description
NBAG	Carney et al. [1]	Neural Bootstrap Aggregation Bagging
Bench	Carney et al. [1]	Benchmark Bagging Ensemble
Simple	Carney et al. [1]	Simple Bagging Ensemble
GRNNFA	Lee et al. [10]	General Regression NN and Fuzzy Reasoning Theory
SONFIN	Juang et al. [6]	Self-constructing neural fuzzy inference network
T2FLS-G	Mendel [11]	Gradient-descent based IT2FIS tuning
SEIT2FNN	Juang et al. [7]	Self-evolving IT2FIS
IT2FNN-SVR(N)	Juang et al. [5]	IT2 Fuzzy-NN-Support-Vector Regression-Numeric Input
IT2FNN-SVR(F)	Juang et al. [5]	IT2 Fuzzy-NN-Support-Vector Regression-Fuzzy Input

IT2FIS training and test results (10 fold CV)

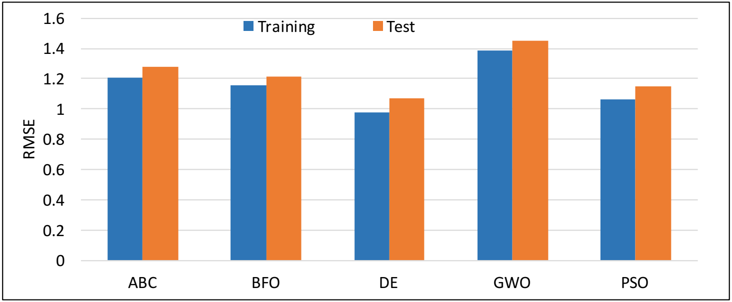
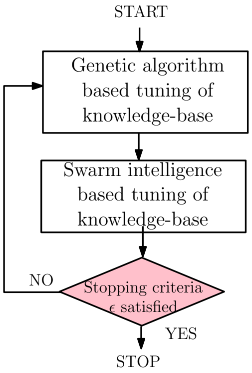
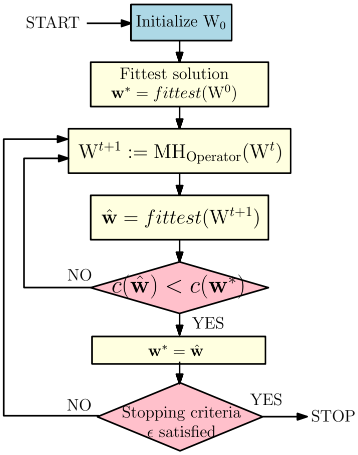


Figure: Average training test performance of algorithms for all datasets

How result was achieved



Knowledge base optimization



Metaheusitic basic framework

Evolutionary algorithm

1. Selection operation

Parent 1	1	0	1	0	1	1	0	1	1	0
Parent 2	0	0	0	1	0	0	1	0	1	1
Parent 3	0	1	0	1	1	0	0	1	1	0
Parent 4	0	0	0	1	1	1	0	1	1	0
Parent 5	1	1	0	1	0	0	0	1	0	0
:										
Parent N	0	0	1	1	1	1	0	1	0	0

2. Crossover operation

Cut point

Parent 1	1	0	1	0	1	1	0	1	1	0
Parent 2	0	0	0	1	1	1	1	1	0	1

Crossover

Child 1	1	0	1	0	1	1	1	1	0	1
Child 2	0	0	0	1	1	1	0	1	1	0

3. Mutation operation

Mutation point

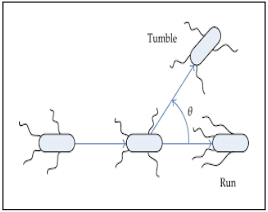
Parent 1	0	0	0	1	1	1	1	1	0	1
Child 1	0	0	0	1	0	1	1	1	0	1

4. Recombination operation

Example of algorithms:

1. Genetic algorithm (Goldberg and Holland [3])
2. Differential evolution (Storn and Price [15])

Swarm Inspired Algorithms



Example of algorithms:

1. Particle swarm optimization (Eberhart and Kennedy [2])
2. Artificial bee colony (Karaboga [8])
3. Bacteria foraging optimization (Passino [14])
4. Gray wolf optimization (Mirjalili et al. [13])

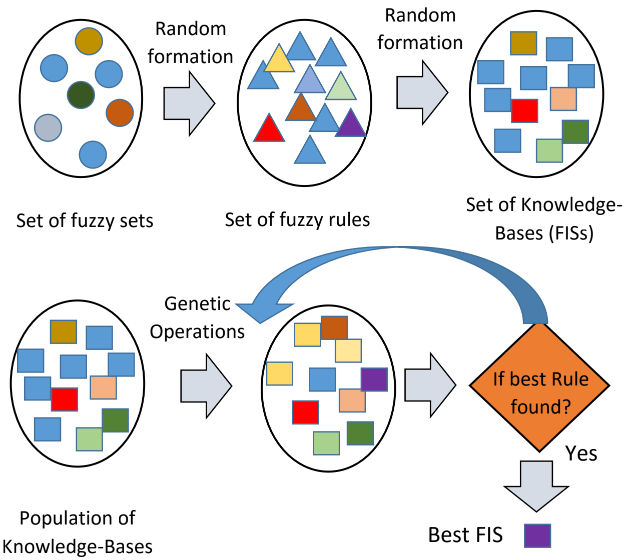
IT2FIS optimization: Pittsburgh approach

1. Each rule in a FIS are randomly assigned a status "0" (inactive) or "1" (active).
2. The i-th rule base (FIS) $\mathbf{R}_i$ is defined as:

$$\mathbf{R}_i = \{R_{i1}, R_{i2}, \ldots, R_{iM}\} \, \forall \, i = 1, 2, \ldots, k, \tag{1}$$

   where the status of a rule $R_{ij} \in \mathbf{R}_i$ is randomly set to either "0" or "1."
3. A population $Q = (\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_k)$ of a total k FISs are generated.
4. Population Q is a genetic population, where the individuals are coded into a binary vector.

Illustration Pittsburgh approach



IT2FIS optimization: Michigan approach

1. A single rule or rule parameters are encoded into a genotype.
2. For a rule R^i that has a total p^i fuzzy set, the parameter vector $\mathbf{w}^i$ is designed as follows:

$$\mathbf{w}^i = \langle (m, \lambda, \sigma)_1^i, \ldots, (m, \lambda, \sigma)_{p^i}^i, (c_0, s_0)^i, (c_1, s_1)^i, \ldots, (c_{p^i}, s_{p^i})^i \rangle$$

   where $(m, \lambda, \sigma)_j^i$ is the parameters (center, center deviation factor, and width) of the j-th T2FS.
3. The pairs $(c_j, s_j)^i$, $j = 0$ to p^i are the consequent part parameters and their deviation factors.

Illustration Michigan approach

Parameter vector of the best obtained FIS:

w_1	w_2	w_2	w_3	w_4	w_5	w_6	…	w_{n-1}	w_N

Differential evolution for the optimization of parameter vector of the best obtained FIS:

Solution 1
Solution 2
Solution 3
Solution 4
:
:
Solution N-2
Solution N-1
Solution N

Gen 1: A population of N Solutions

A new solution is created beaded on a Crossover Factor **CR** and by picnicking-up the following solution vectors from the population:

Best solution | Solution a | Solution b | Solution c

A New Solution = Solution a +

F x (Best solution - Solution a) +

F x (Solution b - Solution c) +

F – Mutation Factor.

New Solution
Solution 2
Solution 3
Solution 4
:
:
Solution N-2
Solution N-1
Solution N

Gen 2: A population of N Solutions

Limitations: Best algorithms identification for the datasets

Table: Training and test performance of the algorithms

Data	Training	Test
Abalone	DE	DE
Diabetes	DE	GWO
Elevator	DE	PSO
Fridman	DE	DE

Limitations: Convergence trajectories FRD dataset

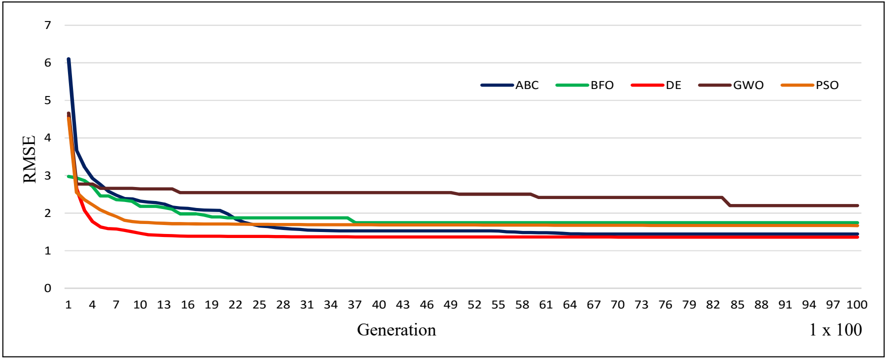


Figure: Dataset FRD

Limitations:
Convergence trajectories

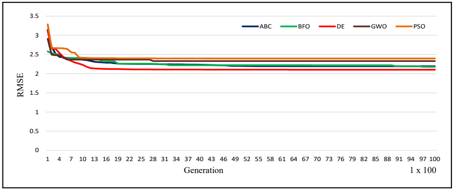


Figure: Dataset ABL

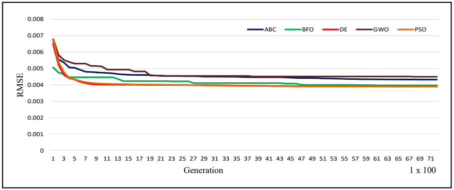


Figure: Dataset ELV

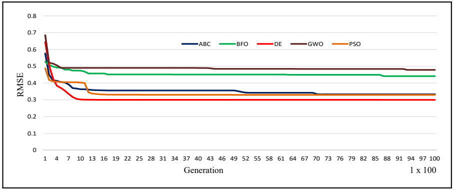


Figure: Dataset DIB

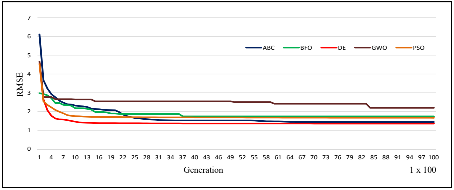


Figure: Dataset FRD

Conclusions

1. Random combination of fuzzy sets for the creation of rules followed by random combination of rules for the creation of fuzzy inference system (FIS), and the genetic operation on the population of FISs helps in obtaining optimum FIS.
2. Metaheuristic for the IT2FIS parameter optimization helps in improving approximation ability of FIS.
3. Use of fixed weights (left and right weights) at the type-2 rules was useful during ,metaheuristic tuning.
4. No one algorithm performed best in all cases. For example, test performance of GWO and PSO was best for the datases DIB and ELV respectively; whereas DE performed best for the others.
5. Convergence trajectory of the algorithms were tended to slow down quickly during IT2FIS optimization.

Future scope

1. Nature inspired heuristic may be used instead of random combination fuzzy sets for constructing rules. Similarly, such heuristic may be used for the construction FISs from the set of rules.
2. Gradient based local search algorithm can be combined with metaheuristic to avoid falling local minima.

Thank You!

varun.kumar.ojha.@vsb.cz

Outline

1. Results: Parameter setting
2. Fuzzy Inference System
   - Interval Type-II FIS
   - Type-II Fuzzy Set
3. References

Parameter setting and datasets

Table: Parameter Setting of the Algorithms

#	Algo.	Pop.	Eval.	Other
1	ABC	50	5000	$t_{abc} = 100$
2	BFO	50	5000	$N_r = N_d = N_s = 10$, $d_r = 0.25$
3	DE	50	5000	CR = 0.9, F = 0.7
4	GWO	50	5000	-
5	PSO	50	5000	$c0 = 0.729$, $c_1 = 1.49$ $c_2 = 1.49$

Table: Datasets for the Experiments

#	Dataset	Abbr.	Attributes	Samples	Difficulty
1	Abalone	ABL	8	4177	130
2	Diabetes	DIB	2	43	40
3	Elevator	ELV	18	16599	280
4	Fridman	FRD	5	1200	85

Note: The dataset FRD has random Gaussian noise included as mentioned in [5]

Interval Type-II Fuzzy Inference System

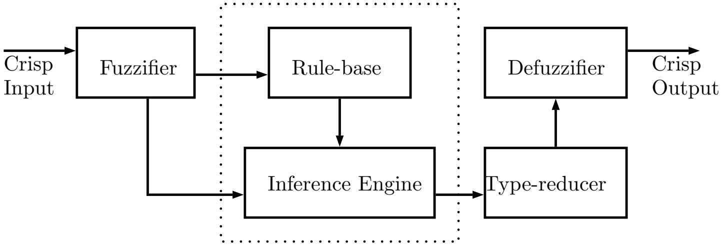


Figure: Type-2 Fuzzy Inference System

A rule in type-2 FIS may be represented as:

$$R^i : \text{IF } x_1 \text{is } \tilde{A}_1^i \text{and } \ldots \text{and } x_n \text{is } \tilde{A}_n^i \text{THEN } y \text{ is } B_n^i \tag{2}$$

where $B = [\bar{b}, \underline{b}]$ is weights at consequent part of the rule and $i = 1, 2, \ldots, M$.

Type-II Fuzzy Set

A T2FS $\tilde{A}$ is characterized by a 3-dimensional membership function [12].

$$\tilde{A} = \left\{ \left((x, u), \mu_{\tilde{A}}(x, u) \right) \middle| \forall x \in X, \forall u \in [0, 1] \right\}. \tag{3}$$

1. x-axis is called primary variable (x).
2. y-axis is called secondary variable or primary MF (u).
3. the value of the T2MF is considered to be along the z-axis.

Type-II Fuzzy Set (Contd..)

A Gaussian function with uncertain mean within $[m_1, m_2]$ and standard deviation σ is an interval type-2 MF:

$$\mu_{\tilde{A}}(x, m, \sigma) = \exp\left[-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)\right], \quad m \in [m_1, m_2]. \tag{4}$$

The upper MF [9]:

$$\bar{\mu}_{\tilde{A}}(x) = \begin{cases} \mu_{\tilde{A}}(x, m_1, \sigma), & x < m_1 \\ 1, & m_1 \leq x \leq m_2 \\ \mu_{\tilde{A}}(x, m_2, \sigma), & x > m_2 \end{cases} \tag{5}$$

and the lower MF [9]:

$$\underline{\mu}_{\tilde{A}}(x) = \begin{cases} \mu_{\tilde{A}}(x, m_2, \sigma), & x \leq (m_1 + m_2)/2 \\ \mu_{\tilde{A}}(x, m_1, \sigma), & x > (m_1 + m_2)/2 \end{cases} \tag{6}$$

Type-II Fuzzy Set (Contd..)

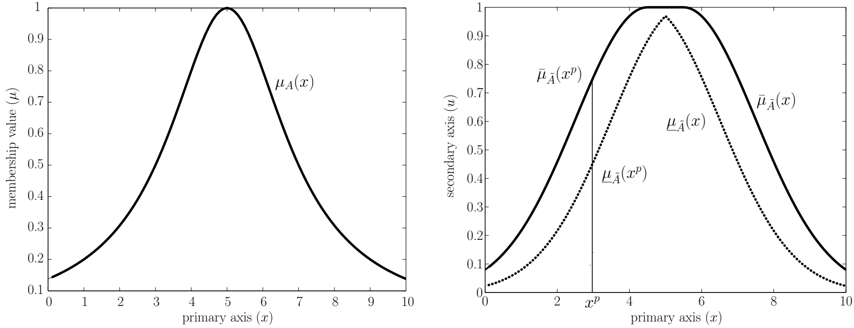


Figure: Type-1 MF with mean $m = 5.0$ (*left*). Type-2 Fuzzy MF with fixed $\sigma = 2.0$ and means $m_1 = 4.5$ and $m_2 = 5.5$. Upper MF in bold line and lower MF in doted line are defined as per (5) and (6) (*right*).

References I

[1] John G Carney and Padraig Cunningham.
Tuning diversity in bagged ensembles.
International Journal of Neural Systems, 10(04):267–279, 2000.

[2] R. Eberhart and J. Kennedy.
A new optimizer using particle swarm theory.
In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995. MHS '95.*, pages 39–43, 1995.

[3] David E Goldberg and John H Holland.
Genetic algorithms and machine learning.
Machine Learning, 3(2):95–99, Oct 1988.

[4] Hisao Ishibuchi, Tomoharu Nakashima, and Tetsuya Kuroda.
A hybrid fuzzy genetics-based machine learning algorithm: hybridization of michigan approach and pittsburgh approach.
In *1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings.*, volume 1, pages 296–301. IEEE, 1999.

[5] Chia-Feng Juang, Ren-Bo Huang, and Wei-Yuan Cheng.
An interval type-2 fuzzy-neural network with support-vector regression for noisy regression problems.
IEEE Transactions on Fuzzy Systems, 18(4):686–699, 2010.

[6] Chia-Feng Juang and Chin-Teng Lin.
An online self-constructing neural fuzzy inference network and its applications.
IEEE Transactions on Fuzzy Systems, 6(1):12–32, 1998.

[7] Chia-Feng Juang and Yu-Wei Tsao.
A self-evolving interval type-2 fuzzy neural network with online structure and parameter learning.
IEEE Transactions on Fuzzy Systems, 16(6):1411–1424, 2008.

References II

[8] Dervis Karaboga.
An idea based on honey bee swarm for numerical optimization.
Technical Report TR06, Computer Engineering Department, Erciyes University, 2005.

[9] Nilesh N Karnik, Jerry M Mendel, and Qilian Liang.
Type-2 fuzzy logic systems.
IEEE Transactions on Fuzzy Systems, 7(6):643–658, 1999.

[10] Eric WM Lee, Chee Peng Lim, Richard KK Yuen, and SM Lo.
A hybrid neural network model for noisy data regression.
IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 34(2):951–960, 2004.

[11] Jerry M Mendel.
Computing derivatives in interval type-2 fuzzy logic systems.
IEEE Transactions on Fuzzy Systems, 12(1):84–98, 2004.

[12] Jerry M Mendel.
On km algorithms for solving type-2 fuzzy set problems.
IEEE Transactions on Fuzzy Systems, 21(3):426–446, 2013.

[13] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis.
Grey wolf optimizer.
Adv. Eng. Softw., 69:46–61, 2014.

[14] Kevin M Passino.
Biomimicry of bacterial foraging for distributed optimization and control.
IEEE Control Syst., 22(3):52–67, Jan 2002.

[15] Rainer Storn and Kenneth Price.
Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces.
J. Glob. Optim., 11(4):341–359, Dec 1997.

References III