Hindawi Publishing Corporation
Applied Computational Intelligence and Soft Computing
Volume 2012, Article ID 347157, 21 pages
doi:10.1155/2012/347157

Research Article

Data and Feature Reduction in Fuzzy Modeling through Particle Swarm Optimization

S. Sakinah S. Ahmad[1] and Witold Pedrycz[1,2]

[1] *Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada T6G 2G7*
[2] *Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland*

Correspondence should be addressed to S. Sakinah S. Ahmad, sh_sakinah@yahoo.com

Received 15 August 2011; Revised 1 November 2011; Accepted 8 December 2011

Academic Editor: Miin-Shen Yang

The study is concerned with data and feature reduction in fuzzy modeling. As these reduction activities are advantageous to fuzzy models in terms of both the effectiveness of their construction and the interpretation of the resulting models, their realization deserves particular attention. The formation of a subset of meaningful features and a subset of essential instances is discussed in the context of fuzzy-rule-based models. In contrast to the existing studies, which are focused predominantly on feature selection (namely, a reduction of the input space), a position advocated here is that a reduction has to involve both data and features to become efficient to the design of fuzzy model. The reduction problem is combinatorial in its nature and, as such, calls for the use of advanced optimization techniques. In this study, we use a technique of particle swarm optimization (PSO) as an optimization vehicle of forming a subset of features and data (instances) to design a fuzzy model. Given the dimensionality of the problem (as the search space involves both features and instances), we discuss a cooperative version of the PSO along with a clustering mechanism of forming a partition of the overall search space. Finally, a series of numeric experiments using several machine learning data sets is presented.

1. Introduction

In fuzzy modeling, the two main approaches for generating the rules rely on knowledge acquisition from human experts and knowledge discovery from data [1, 2]. In recent years, knowledge discovery from data or data-driven fuzzy modeling has become more important [2–4]. In many cases, the ability to develop models efficiently is hampered by the dimensionality of the input space as well as the number of data. If we are concerned with rule-based models, the high-dimensionality of the feature space along with the topology of the rules gives rise to the curse of dimensionality [1, 4]. The number of rules increases exponentially and is equal to P^n, where n is the number of features and P stands for the number of fuzzy sets defined for each feature.

The factors that contribute most to the accuracy of the data-driven fuzzy modeling are associated with the size of the input space and the decomposition of the input data. A Large number of data points or instances in a continuous input-output domain exhibit a significant impact on fuzzy models. It is well known that more training data will not always lead to a better performance for data-driven models. Large amount of training data have important implications on the modeling capabilities. Since the number of fuzzy sets determines the family of realizable approximation functions, larger datasets present the possibility of over-fitting the training data [1, 4]. Thus, the effectiveness of the fuzzy models relies on the quality of the training data. In addition, the main drawback is the fuzzy models' relative inefficiency as the size of the data increases, regarding both the number of data points in the data set and the number of features. Moreover, one of the most widely used approaches in fuzzy modeling is the fuzzy C-means (FCM) algorithm for constructing the antecedents of the rules associated with the curse of dimensionality [5, 6].

The dimensionality problem can be addressed by reducing the constructed fuzzy rules. The reduction method plays two important roles: it increases the effectiveness of

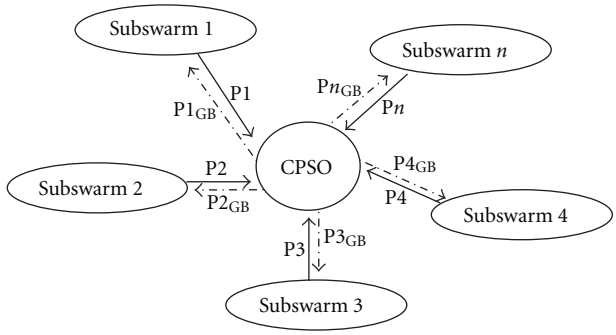

Figure 1: The schematic diagram of information sharing in CPSO.

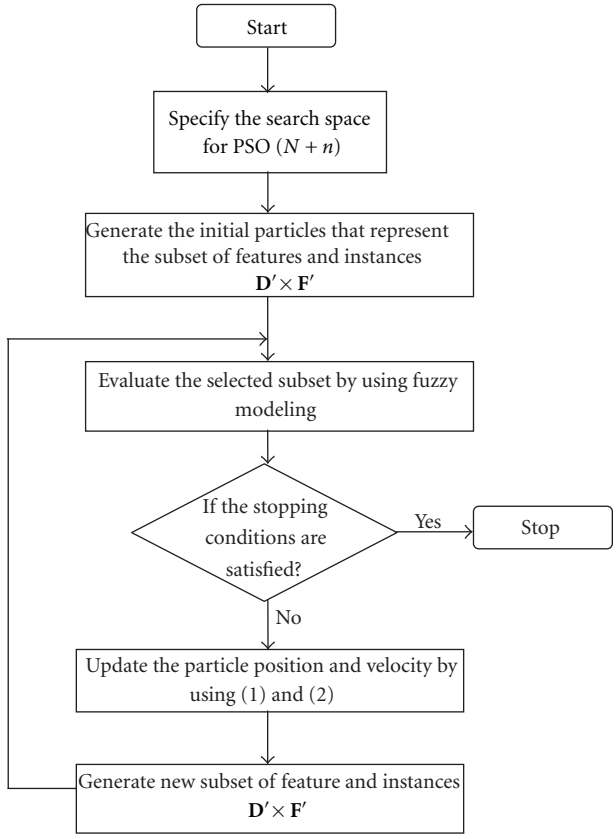

Figure 2: The scheme of the proposed data and reduction for fuzzy modeling.

the learning algorithm, since the learning algorithm will concentrate only on the most useful subset of data, and it also improves the computational efficiency as the learning algorithm involves only a subset of data smaller than the original dataset [7]. This reduction can be realized by removing the redundant fuzzy rules by exploiting a concept of fuzzy similarity [3, 7, 8]. Evolutionary algorithms have also been used for building compact fuzzy rules [9–12]. An evolutionary algorithm is used to tune the structure and the rules' parameter of the fuzzy systems [13, 14]. However, in numerous cases, some variables are not crucial to the realization of the fuzzy model. A suitable way to overcome this problem is to implement feature selection before constructing the fuzzy models. Therefore, during the last decade, feature selection methods in conjunction with constructing fuzzy models for reducing the curse of dimensionality were developed [15–22]. This process reduces the fuzzy rule search space and increases the accuracy of the model.

As mentioned above, forming the best input data as the training set to construct the fuzzy modeling is also important. However, as far as we know there is no research that has been done to simultaneously select the best subset of features and input data for constructing the fuzzy model. Most of the research is focused on reducing the fuzzy rules, and the process of simplifying the system is done once the design has been completed. Here we propose a method that reduces the complexity of the system starting from the design stage. however, the process of constructing the antecedent and the consequent parts of the fuzzy model is realized using the best subset of input data.

In this paper, a comprehensive framework is proposed to construct fuzzy models from the subset of numerical input-output data. First, we develop a data-driven fuzzy modeling framework for a high-dimensional large dataset, which is capable of generating a rule-based automatically from numerical data. Second, we integrate the concept of feature selection and data selection together in the unified form to further refine (reduce) the fuzzy models. In this regard, the PSO technique is applied in order to search for the best subset of data. In order to increase the effectiveness of the PSO techniques, we introduce a new cooperative PSO method based on the information granulation approach. Third, we develop a flexible setup to cope with the optimization of variables and data to be used in the design of the fuzzy model. The proposed approach allows the user to choose the predetermined fraction of variables and data that can be used to construct the fuzzy models.

This paper is organized as follows. We briefly elaborate on the selected approaches to data and feature space reduction in Section 2, and then in Section 3, we recall the main algorithmic features of PSO and its cooperative version, CPSO, which is of interest in problems of high-dimensionality. The proposed fuzzy modeling framework along with its main algorithmic developments is presented in Section 4. Experimental studies are presented in Section 5, and conclusions are provided in Section 6.

2. Selected Approaches to Data and Space Reduction

In general, reduction processes involve feature selection (FS), instances (data) selection (IS), and a combination of these two reduction processes: feature and Instances selection (FIS). Feature selection is a subject of the main reduction pursuits. The goal of FS, which is commonly encountered in problems of system modeling and pattern recognition, is to select the best subset of features so that the model formed in this new feature (input) space exhibits the highest accuracy (classification rate) being simultaneously associated with the

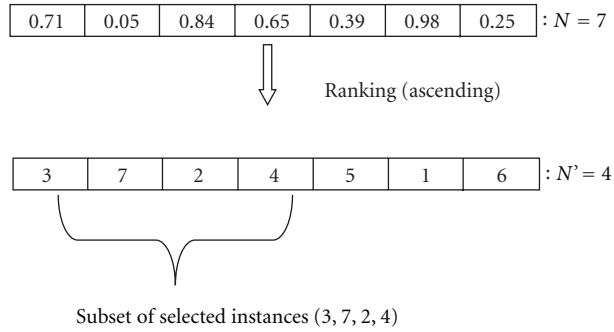

Figure 3: From a particle in $[0, 1]^{N+n}$ search space to a subset of instances and features.

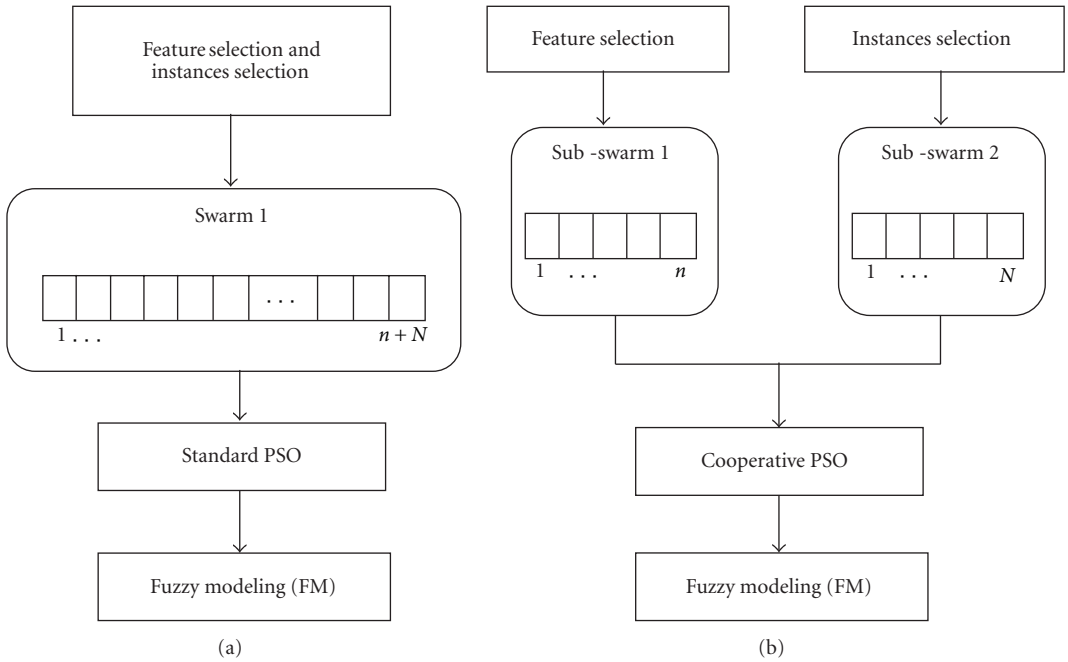

Figure 4: The particle scheme of the "standard" PSO (a) and cooperative PSO (b).

increased transparency of the resulting construct [23]. The process aims to discard irrelevant and/or redundant features [24]. In general, the FS algorithms can be classified into three main categories: filters, wrappers, and embedded methods. The filter method selection criterion is independent of the learning algorithm. In contrast to the wrapper method, the selection criterion is dependent on the learning algorithm and uses its performance index as the evaluation criterion. The embedded method incorporates feature selection as part of the training process. The reader can refer to [23–25] for more details.

Instances selection (IS), another category of reduction approaches, is concerned with the selection of the relevant data (instances) reflective of the knowledge pertinent to the problem at hand [26, 27]. The three main functions forming the essence of IS include enabling, focusing and cleaning [26].

In this study, as stated earlier, instead of approaching feature selection and instances selection separately, we focus on the integration of feature selection and instances selection in the construction of the fuzzy models. Both processes are applied simultaneously to the initial dataset, in order to obtain a suitable subset of feature and data to construct the parameters for the fuzzy model. In the literature, some methods for integrating feature and instances selection are more focused on a class of classification problems [28, 29].

The ideas of feature and data reduction as well as hybrid approaches have been discussed in the realm of fuzzy modeling. Table 1 offers a snapshot at the diversity of the existing approaches and the advantages gained by completing the reduction processes.

3. Particle Swarm Optimization and Its Cooperative Version

Population-based algorithms provide interesting solutions since any constructive method can be used to generate the initial population, and any local search technique can

Table 1: A summary of selected studies in data and feature reduction in fuzzy modeling.

Reference	Feature reduction technique	Dataset, fuzzy model and data	Original data used in modeling		Number of selected features	Number of resulting rules
			Number of instances	Number of features		
Gaweda et al. [15]	The use of sensitivity analysis Determination of essential features	Box-Jenkins gas furnace	296	10	3	2
Hadjili and Wertz [16]	Deviation criterion (DC): to measure the change in fuzzy partition. Removal of features that do not significantly change the fuzzy partition	Nonlinear systems in noisy environment	250	3	1	4
		Nonlinear dynamical system excited by a sinusoidal signal	800	10	6	8
		Run-out cooling table in a hot strip mill	1000	17	5	12
Zarandi et al. [18]	Heuristic method to select features	Nonlinear System used in [3]	50	4	2	4
		Supplier chance management dataset	300	9	5	5
Du and Zhang [19]	Evolutionary optimization	Box-Jenkins gas furnace	296	10	3	4
		MR damper identification	5000	11	6	10
Ghazavi and Liao [20]	(1) Mutual correlation methods, (2) gene selection criteria (3) the relief algorithm	Wisconsin breast cancer	569	30	3	250 (3)
		PIMA Indian diabetes	768	8	3	125 (3)
		Welding flaw identification	399	25	3	—
Zhang et al. [21]	Iterative search margin based algorithm (Simba)	Wisconsin breast cancer	699	9	5	3
		Wine	178	13	4	5
		Iris	150	4	3	3
		Ionosphere	351	34	10	4

Table 2: Description of data used in the experiments.

Data set	Abbreviation	Number of features	Number of data	Sparsity ration, κ
Air pollution PM10	PM10	7	500	71.43
Boston housing	Housing	13	506	38.92
Body fat	Body fat	14	252	18.00
Parkinson's telemonitoring	Parkinson	17	5875	345.59
Computer activity	Computer	21	8192	390.09

be used to improve each solution in the population [30]. In addition, population-based methods have the advantage of being able to combine good solutions in order to obtain potentially better ones. Most of the population-based algorithm approaches in FS and IS are based on GAs. Some recent studies [28, 29, 31] have employed population-based optimization techniques to carry out search for the best subset of variables and data for solving the application problems, but all of them were carried out to solve the classification problem. Therefore, in this study, we use population-based technique for selecting the best subset of feature and data for the regression problem. Here, we implement particle swarm optimization (PSO) techniques to intelligently search for the best subset of features and data (instances).

PSO, developed by Kennedy and Eberhart, inspired by the collective behavior of birds or fish [32], is a population-based algorithm where each individual, referred to as a particle, represents a candidate solution. Each particle proceeds through the search space at a given velocity v that

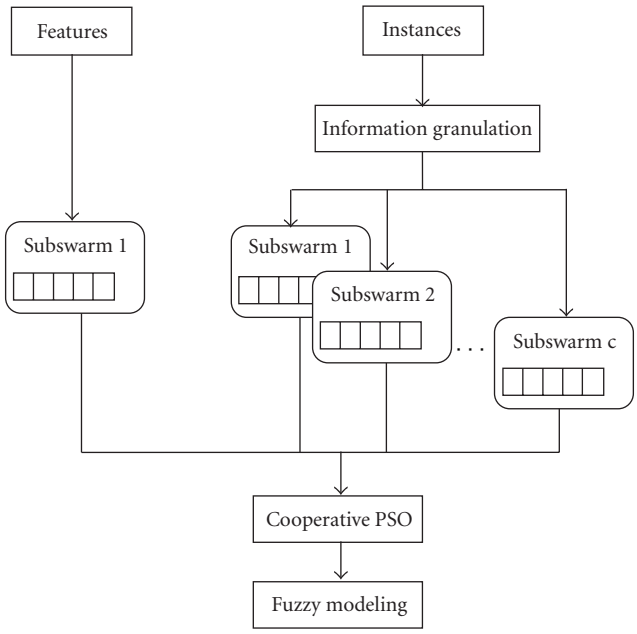

Figure 5: The particle scheme of cooperative PSO with more subswarms.

Table 3: The values of the parameters used in the experiments; CPSO[1]: swarms located in the feature space. CPSO[2]: swarms located in the instance (data) space.

Optimization method	Subswarms	Generation	Particles
PSO	1	50	300
CPSO[1]	2	50	100
CPSO[2]	4	30	50

is dynamically modified according to its own experience and results in its local best (lb) performance. It is also affected by others particles flying experience resulting in the best value, global best (gb). The underlying expression for the update of the velocity in successive generations reads as follows:

$$v_i\ (t+1) = w \cdot v_i(t) + c_1 \cdot r_{1,i}(t)[\mathrm{lb}_i(t) - x_i(t)] + c_2 \cdot r_{2,i}(t)[\mathrm{gb}_i(t) - x_i(t)], \quad (1)$$
$$x_i(t+1) = x_i(t) + v_i(t+1),$$

where $i = 1, 2, \ldots, N + n$ (the search space is equal to the sum of the dimensionalities of the feature space and the size of the data). The inertia weight (w) is confined to the range [0, 1]; its values can decrease over time. The cognitive factor c_1 and social factor c_2 determine the relative impact coming from the particle's own experience and the local best and global best. r_1 and r_2 are numbers drawn from a uniform distribution over the unit interval that brings some component of randomness to the search process.

In this research, we employed the PSO-based method to handle two optimization tasks, namely, (1) selection of the optimal subset of features and (2) selection of the optimal subset of instances based on the concept of information granularity. In order to reduce the computational complexity of using the standard PSO, we employed cooperative PSO method to simultaneously solve the two optimization tasks. The motivation behind the use of cooperative PSO, as advocated in [33], is to deal effectively with the dimensionality of the search space, which becomes a serious concern when a large number of data with a large dimensionality are involved. This curse of dimensionality is a significant impediment negatively impacting the effectiveness of standard PSO. The essence of the cooperative version of PSO is essentially a parallel search for optimal subset of features and its optimal subset of instances. The cooperative strategy is achieved by dividing the candidate solution vector into components, called subswarm, where each subswarm represents a small part of the overall optimization processes. By doing this, we implement the concept of divide and conquer to solve the optimization problem, so that the process will become more efficient and fast.

The mechanism of information sharing of CPSO is shown in Figure 1. The cooperative search between one subswarm and other is achieved by sharing the information of the global best position (P_{GB}) across all subswarm. Here the algorithm has the advantage of taking two steps forward because the candidate solution comes from the best position for all subswarm except only for the current subswarms being evaluated. Therefore, the algorithm will not spend too much time optimizing the features or instances that have little effect to the overall solution. The rate at which each swarm converges to the solution is significantly higher than the rate of convergence of the standard PSO.

The essence of the cooperative version of PSO is to split the data into several groups so that each group is handled by a separate PSO. The main design question involves splitting the variables into groups. A sound guideline is to keep the related (associated) variables within the same group. Obviously, such relationships are not known in advance. Several possible methods are available for addressing this issue in more detail in the context of the problem at hand.

(a) As we are concerned with a collection of features and data (instances), a natural way to split the variables would be to form two groups ($K = 2$), one for the features (n) and another one for the instances (N). This split would be legitimate if the dimensionality of both subsets was quite similar.

(b) In some situations, one of the subsets (either the data or the features) might be significantly larger than the other one. We often encounter a large number of data, but in some situations, a large number of features might be present (for instance, in microarray data analysis). This particular collection of data or features is then split into K groups. Clustering such items is a viable algorithmic approach. Running K-means or fuzzy C-means produces clusters (group) of variables that are used in the individual PSO.

(c) In case both subsets are large, the clustering is realized both for the features and data, and the resulting, structure (partition) is used to run cooperative PSO.

Table 4: Results for housing data; the number of clusters is set to 4, $c = 4$; κ is the ratio of the number of selected data versus the number of selected features.

Feature	Data = 10% (# of data = 51)		Data = 20% (# of data = 101)		Data = 30% (# of data = 152)		Data = 40% (# of data = 202)		Data = 50% (# of data = 253)		Data = 60% (# of data = 304)		Data = 70% (# of data = 354)		Data = 80% (# of data = 405)		Data = 90% (# of data = 455)		Data = 100% (# of data = 506)	
	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE
10%	51.0	6.341 ± 0.253	101.0	6.262 ± 0.171	152.0	5.777 ± 0.207	202.0	6.227 ± 0.351	253.0	6.389 ± 0.063	304	6.483 ± 0.214	354.0	6.610 ± 0.211	405.0	6.387 ± 0.026	455.0	6.655 ± 0.061	506	7.437 ± 0
20%	17.0	6.664 ± 0.233	33.7	5.389 ± 0.253	50.7	5.191 ± 0.185	67.3	4.884 ± 0.118	84.3	4.805 ± 0.047	101	4.882 ± 0.080	118.0	4.906 ± 0.040	135.0	5.011 ± 0.077	151.7	5.149 ± 0.071	169	5.039 ± 0.096
30%	12.8	6.127 ± 0.245	25.3	5.468 ± 0.290	38.0	4.853 ± 0.183	50.5	4.574 ± 0.078	63.3	4.619 ± 0.294	76	4.45 ± 0.092	88.5	4.398 ± 0.035	101.3	4.56 ± 0.053	113.8	4.536 ± 0.026	127	4.578 ± 0.01
40%	10.2	6.321 ± 0.518	20.2	5.122 ± 0.245	30.4	4.626 ± 0.075	40.4	4.362 ± 0.190	50.6	4.172 ± 0.123	60.8	4.06 ± 0.099	70.8	4.126 ± 0.105	81.0	4.18 ± 0.177	91.0	4.343 ± 0.026	101	4.233 ± 0.079
50%	7.3	7.126 ± 0.835	14.4	5.046 ± 0.312	21.7	4.574 ± 0.206	28.9	4.018 ± 0.109	36.1	3.916 ± 0.224	43.4	3.927 ± 0.089	50.6	4.009 ± 0.093	57.9	4.085 ± 0.132	65.0	4.077 ± 0.092	72.3	4.005 ± 0.082
60%	6.4	8.133 ± 0.782	12.6	5.120 ± 0.189	19.0	4.504 ± 0.207	25.3	4.052 ± 0.196	31.6	3.912 ± 0.120	38	3.935 ± 0.129	44.3	3.934 ± 0.035	50.6	3.923 ± 0.117	56.9	3.931 ± 0.102	63.3	3.803 ± 0.088
70%	5.7	9.379 ± 0.984	11.2	5.003 ± 0.232	16.9	4.345 ± 0.134	22.4	3.949 ± 0.125	28.1	3.721 ± 0.071	33.8	3.722 ± 0.065	39.3	3.722 ± 0.066	45.0	3.787 ± 0.046	50.6	3.799 ± 0.041	56.2	3.668 ± 0.043
80%	5.1	10.57 ± 2.251	10.1	5.107 ± 0.262	15.2	4.232 ± 0.173	20.2	3.67 ± 0.093	25.3	3.617 ± 0.108	30.4	3.659 ± 0.128	35.4	3.568 ± 0.064	40.5	3.567 ± 0.086	45.5	3.645 ± 0.065	50.6	3.526 ± 0.049
90%	4.3	24.05 ± 7.681	8.4	5.324 ± 0.207	12.7	4.173 ± 0.181	16.8	3.809 ± 0.080	21.1	3.652 ± 0.050	25.3	3.569 ± 0.028	29.5	3.555 ± 0.017	33.8	3.533 ± 0.016	37.9	3.541 ± 0.015	42.2	3.550 ± 0
100%	3.9	44.39 ± 17.65	7.8	5.409 ± 0.201	11.7	4.082 ± 0.047	15.5	3.781 ± 0.055	19.5	3.687 ± 0.038	23.4	3.654 ± 0.015	27.2	3.631 ± 0.021	31.2	3.615 ± 0.011	35.0	3.605 ± 0.015	38.9	4.023 ± 0

Table 5: Results for PM10 dataset; $c = 3$.

Feature	Data = 10% (# of data = 50)		Data = 20% (# of data = 100)		Data = 30% (# of data = 150)		Data = 40% (# of data = 200)		Data = 50% (# of data = 250)		Data = 60% (# of data = 300)		Data = 70% (# of data = 350)		Data = 80% (# of data = 400)		Data = 90% (# of data = 450)		Data = 100% (# of data = 500)	
	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE
10%	50.0	0.931 ± 0.036	100.0	0.979 ± 0.018	150.0	0.983 ± 0.006	200.0	0.985 ± 0.010	250.0	0.998 ± 0.021	300.0	1.036 ± 0.022	350.0	1.071 ± 0.016	400.0	1.088 ± 0.002	450.0	1.099 ± 0.003	500.0	1.116 ± 0
20%	50.0	0.896 ± 0.034	100.0	0.98 ± 0.013	150.0	0.987 ± 0.009	200.0	0.994 ± 0.008	250.0	1.004 ± 0.018	300.0	1.025 ± 0.032	350.0	1.075 ± 0.009	400.0	1.090 ± 0.003	450.0	1.098 ± 0.003	500.0	1.116 ± 0
30%	25.0	0.825 ± 0.087	50.0	0.902 ± 0.04	75.0	0.918 ± 0.007	100.0	0.916 ± 0.003	125.0	0.918 ± 0.010	150.0	0.920 ± 0.005	175.0	0.922 ± 0.003	200.0	0.937 ± 0.001	225.0	0.948 ± 0.005	250.0	0.964 ± 0
40%	16.7	0.829 ± 0.023	33.3	0.877 ± 0.007	50.0	0.877 ± 0.010	66.7	0.862 ± 0.009	83.3	0.865 ± 0.004	100.0	0.869 ± 0.008	116.7	0.887 ± 0.004	133.3	0.892 ± 0.006	150.0	0.902 ± 0.002	166.7	0.915 ± 0
50%	12.5	0.802 ± 0.029	25.0	0.816 ± 0.008	37.5	0.822 ± 0.012	50.0	0.826 ± 0.028	62.5	0.843 ± 0.024	75.0	0.870 ± 0.023	87.5	0.891 ± 0.022	100.0	0.906 ± 0.004	112.5	0.905 ± 0.002	125.0	0.925 ± 0
60%	12.5	0.804 ± 0.027	25.0	0.818 ± 0.013	37.5	0.825 ± 0.014	50.0	0.834 ± 0.016	62.5	0.841 ± 0.006	75.0	0.879 ± 0.012	87.5	0.898 ± 0.006	100.0	0.897 ± 0.006	112.5	0.907 ± 0.002	125.0	0.925 ± 0
70%	10.0	0.783 ± 0.030	20.0	0.781 ± 0.031	30.0	0.804 ± 0.017	40.0	0.782 ± 0.039	50.0	0.806 ± 0.014	60.0	0.832 ± 0.016	70.0	0.851 ± 0.010	80.0	0.856 ± 0.004	90.0	0.864 ± 0.003	100.0	0.900 ± 0
80%	8.3	0.768 ± 0.024	16.7	0.769 ± 0.007	25.0	0.776 ± 0.017	33.3	0.768 ± 0.024	41.7	0.796 ± 0.010	50.0	0.805 ± 0.016	58.3	0.826 ± 0.013	66.7	0.839 ± 0.007	75.0	0.847 ± 0.001	83.3	0.878 ± 0
90%	8.3	0.774 ± 0.026	16.7	0.771 ± 0.017	25.0	0.767 ± 0.014	33.3	0.796 ± 0.010	41.7	0.777 ± 0.019	50.0	0.815 ± 0.017	58.3	0.820 ± 0.014	66.7	0.843 ± 0.001	75.0	0.851 ± 0.003	83.3	0.878 ± 0
100%	7.1	0.786 ± 0.012	14.3	0.758 ± 0.017	21.4	0.764 ± 0.007	28.6	0.765 ± 0.015	35.7	0.772 ± 0.016	42.9	0.795 ± 0.004	50.0	0.808 ± 0.010	57.1	0.818 ± 0.005	64.3	0.824 ± 0.003	71.4	0.883 ± 0

Table 6: Results for Parkinson's data; $c = 3$.

Feature	Data = 10% (# of data = 346)		Data = 20% (# of data = 691)		Data = 30% (# of data = 1037)		Data = 40% (# of data = 1382)		Data = 50% (# of data = 1728)		Data = 60% (# of data = 2074)		Data = 70% (# of data =2419)		Data = 80% (# of data = 2765)	
	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE	κ	RMSE
10%	346	6.388 ± 0.182	691	6.221 ± 0.064	1037	6.393 ± 0.140	1382	6.495 ± 0.106	1728	6.644 ± 0.137	2074	6.644 ± 0.137	2419	6.515 ± 0.006	2765	6.406 ± 0.005
20%	173	6.183 ± 0.110	346	5.857 ± 0.031	519	5.932 ± 0.023	691	5.972 ± 0.008	864	6.247 ± 0.137	1037	6.247 ± 0.137	1210	6.066 ± 0.018	1382	5.970 ± 0.002
30%	115	6.152 ± 0.151	230	5.799 ± 0.061	346	5.703 ± 0.067	461	5.708 ± 0.045	576	6.077 ± 0.048	691	6.077 ± 0.048	806	5.964 ± 0.276	922	5.740 ± 0.089
40%	86	6.328 ± 0.456	173	5.900 ± 0.087	259	5.886 ± 0.342	346	6.175 ± 0.358	432	6.461 ± 0.261	518	6.461 ± 0.261	605	6.299 ± 0.563	691	6.308 ± 0.677
50%	69	6.991 ± 0.525	138	7.585 ± 0.755	207	6.448 ± 0.580	276	7.493 ± 0.303	346	8.057 ± 0.037	415	8.057 ± 0.037	484	8.160 ± 0.018	553	8.110 ± 0.012
60%	58	8.357 ± 0.164	115	8.088 ± 0.028	173	8.069 ± 0.078	230	7.960 ± 0.021	288	8.104 ± 0.021	346	8.104 ± 0.021	403	8.147 ± 0.017	461	8.101 ± 0.002
70%	49	8.401 ± 0.068	99	8.074 ± 0.064	148	8.087 ± 0.015	197	8.008 ± 0.011	247	8.123 ± 0.009	296	8.123 ± 0.009	346	8.158 ± 0.011	395	8.107 ± 0.007
80%	43	8.419 ± 0.091	86	8.257 ± 0.048	130	8.187 ± 0.017	173	8.092 ± 0.020	216	8.177 ± 0.008	259	8.177 ± 0.008	302	8.194 ± 0.008	346	8.138 ± 0.005
90%	38	8.500 ± 0.106	77	8.258 ± 0.024	115	8.199 ± 0.017	154	8.139 ± 0.013	192	8.200 ± 0.009	230	8.200 ± 0.009	269	8.217 ± 0.004	307	8.157 ± 0.006
100%	35	8.560 ± 0.026	69	8.249 ± 0.033	104	8.262 ± 0.013	138	8.223 ± 0.007	173	8.239 ± 0.006	207	8.239 ± 0.006	242	8.235 ± 0.002	276	8.223 ± 0.001

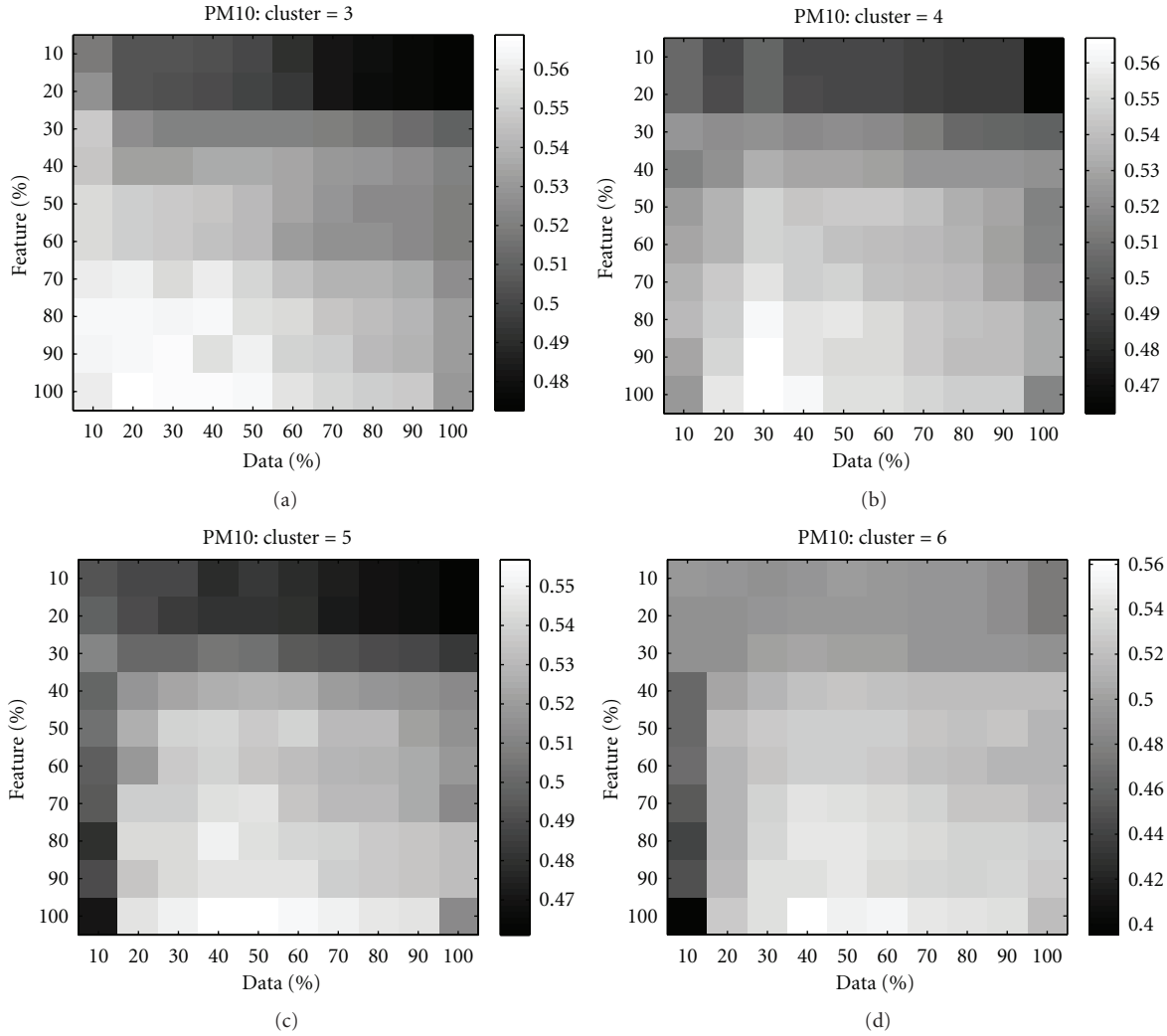

Figure 6: Heat map for PM 10 data for c varying in-between 3 to 6.

Algorithm 1 presents the Cooperative PSO pseudocode implementing the optimization process [33]. Firstly, the PSO is divided into m subspaces, called subswarms. In our case the first subswarm represents the features search space and the rest are for instances search space. $P_j(x_i)$ refers to the position of particle i of subswarms j. The global best for each subswarm defined as $P_{j(\mathrm{GB})}$, and the local best is defined as $P_{j(\mathrm{LB}_i)}$. The cooperation between the subswarms employed in the function $\mathbf{C}(j,k)$, which returns m-dimensional vector formed by concatenating all the global best vector across all subswarms, except for the current position j. Here the jth component is called k and represent the position of any particle from subswarm P_j.

4. PSO-Integrated Feature and Data Reduction in Fuzzy-Rule-Based Models

As the problem of feature data reduction is inherently combinatorial nature, PSO provides an interesting and computationally viable optimization alternative. In the following subsections, we start with a general optimization setting and then discuss the PSO realization of the search process (here, a crucial design phase is a formation of the search space with a suitable encoding mechanism). Although the proposed methodology is of general nature, we concentrate on rule-based models, which are commonly present in fuzzy modeling, to help offer a detailed view of the overall design process.

4.1. An Overall Reduction Process. As is usual in system modeling, we consider a supervised learning scenario ion which we encounter in a finite set of training data $(\mathbf{x}_k, t_k)$, $k = 1, 2, \ldots, N$. By stressing the nature of the data and their dimensionality, the data space along with n-dimensional feature vectors can be viewed as a Cartesian product of the data and features $\mathbf{D} \times \mathbf{F}$. The essence of the reduction is to arrive at the Cartesian product of the reduced data and feature spaces, $\mathbf{D}' \times \mathbf{F}'$, where, $\mathbf{D}' \in \mathbf{D}$ and $\mathbf{F}' \in \mathbf{F}$. The cardinality of the reduced spaces is equal to N' and n', where $N' < N$ and $n' < n$.

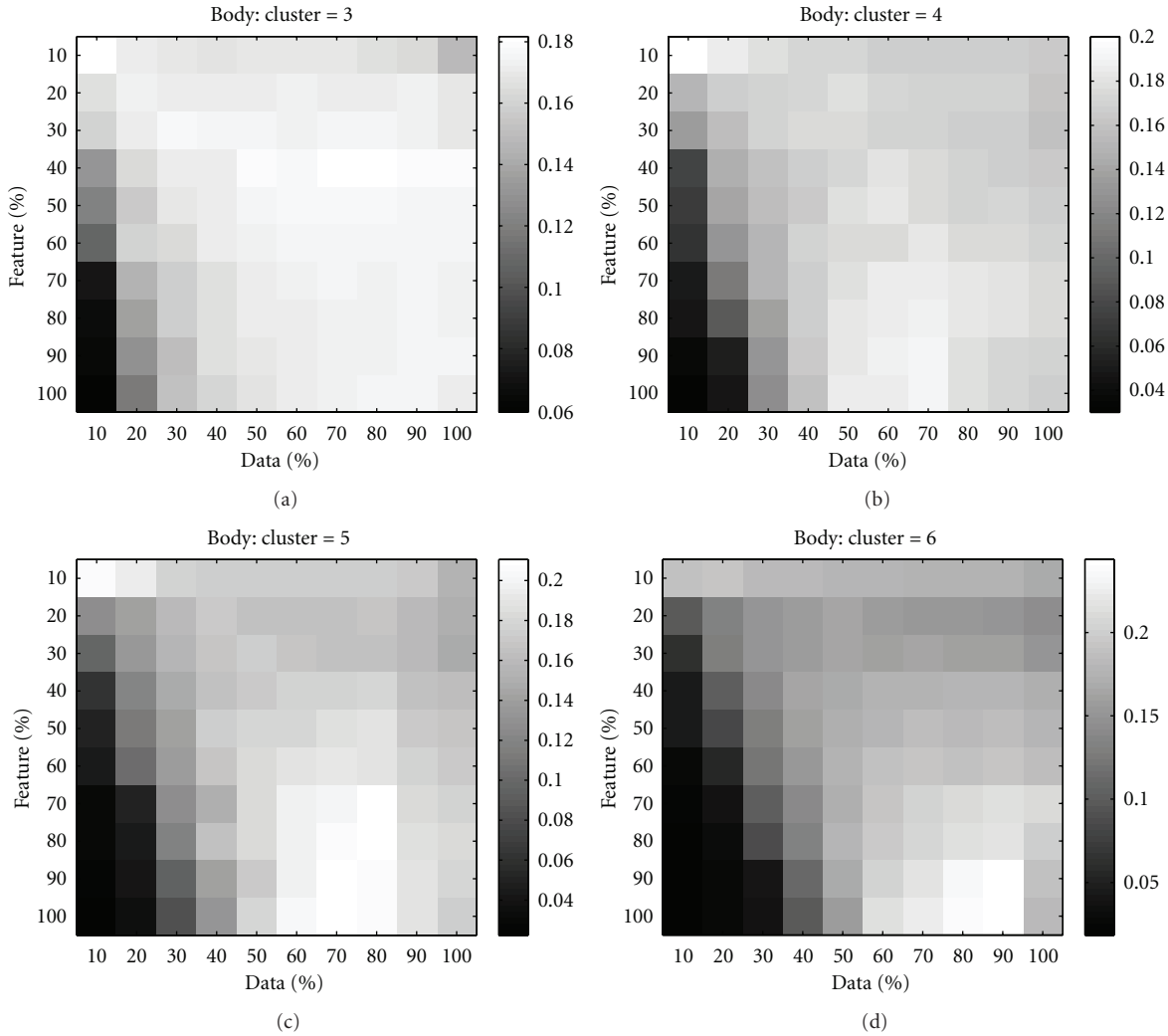

Figure 7: Heat map for Body fat data for varying in-between 3 to 6.

The overall scheme of the reduction process outlining a role of the PSO-guided reduction is illustrated in Figure 2. The scheme can be divided into two important parts and can be described as follows.

(a) Reduction process via PSO: a reduction process tackles both feature reduction and data reduction simultaneously. PSO algorithm is used to search for the best feature and data for constructing the fuzzy model. Here, the size of the selected features (n') and data (N') is provided in advance by the user. After the PSO meets the maximum generation, the process is stopped, and the last best subset of features and data is the best subset of data for constructing the fuzzy model.

(b) Evaluation process: the fuzzy C-means algorithm is used to convert the numerical data into the information granules. Here, the information granularity process deals only with the subset of the data and features ($\mathbf{D}' \times \mathbf{F}'$). Next, the consequent parameter $\mathbf{a}$ constructed from the fuzzy models is use to evaluate the performance of the selected data and features. At this stage we access the performance of the constructed fuzzy model in terms of their capability to fit the model by using the all instances in the original data set.

As it becomes apparent, the original space $\mathbf{D} \times \mathbf{F}$ is reduced, and in this Cartesian product a fuzzy model, denoted by FM, is designed in the usual way (we elaborate on the form of the fuzzy model in the subsequent section). Its design is guided by a certain objective function Q expressed over all elements of original instances. The quality of the reduced space is assessed by quantifying the performance of the fuzzy model operating over the original, non-reduced space. The same performance index as used in the construction of the fuzzy model in the reduced space is used to describe the quality of the fuzzy model:

$$Q = \sqrt{\frac{1}{N} \sum_{\mathbf{x}_k \in \mathbf{D} \times \mathbf{F}'} (\mathrm{FM}(\mathbf{x}_k) - t_k)^2}. \tag{2}$$

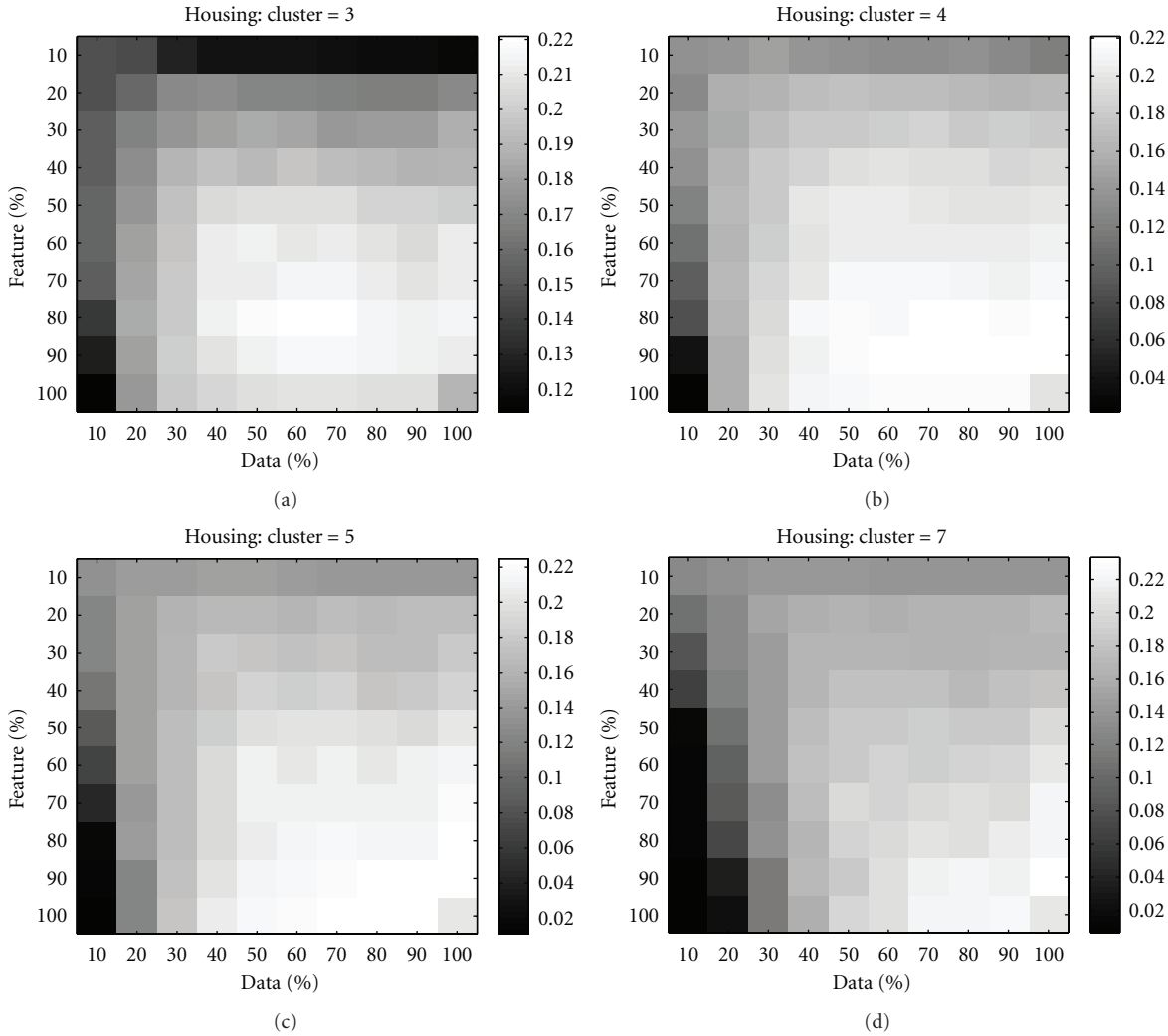

Figure 8: Heat map for housing data for $c = 3$, 4, 5, and 7.

Note that the summation shown above is taken over all the elements forming the data space $\mathbf{D}$. By taking another look at the overall reduction scheme, it is worth noting that the reduction is realized as in the wrapper mode, in which we use a fuzzy model to evaluate the quality of the reduction mechanism.

4.2. The PSO-Based Representation of the Search Space. The reduction of the data and feature spaces involves a selection of a subset of the data and a subset of the features. Therefore, the problem is combinatorial in its nature. PSO is used here to form a subset of integers which are indexes of the data or features to be used in the formation of $\mathbf{D}' \times \mathbf{F}'$. For instance, $\mathbf{D}'$ is represented as a set of indexes $\{i_1, i_2, \ldots, i_{N'}\}$ being a subset of integers $\{1, 2, \ldots, N\}$. From the perspective of the PSO, the particle is formed as a string of real numbers in $[0, 1]$ of the length of $N + n$; effectively, the search space is a hypercube $[0, 1]^{N+n}$. The first substring of length N represents the data; the second one (having n entries) is used to optimize the subset of features. The particle is decoded as follows. Each substring is processed (decoded) separately. The real number entries are ranked. The result is a list of integers viewed as the indexes of the data. The first N' entries out of the N-position substring are selected to form $\mathbf{D}'$. The same process is applied to the substring representing the set of features. An overall decoding scheme is illustrated in Figure 3.

The information given by the PSO is used to represent the subset of features and data to construct the data-driven fuzzy models. Then, the numerical data are represented in terms of a collection of information granules (a fuzzy sets) produced through some clustering (fuzzy clustering). The information about the granules (clusters) is then used to construct the fuzzy models.

In the cooperative PSO, the formation of the search space is realized in a more sophisticated way. The cooperative facet involves mainly exchanging information about the best positions found by the different subswarms. Here, we present a new cooperative PSO (CPSO) algorithm for the

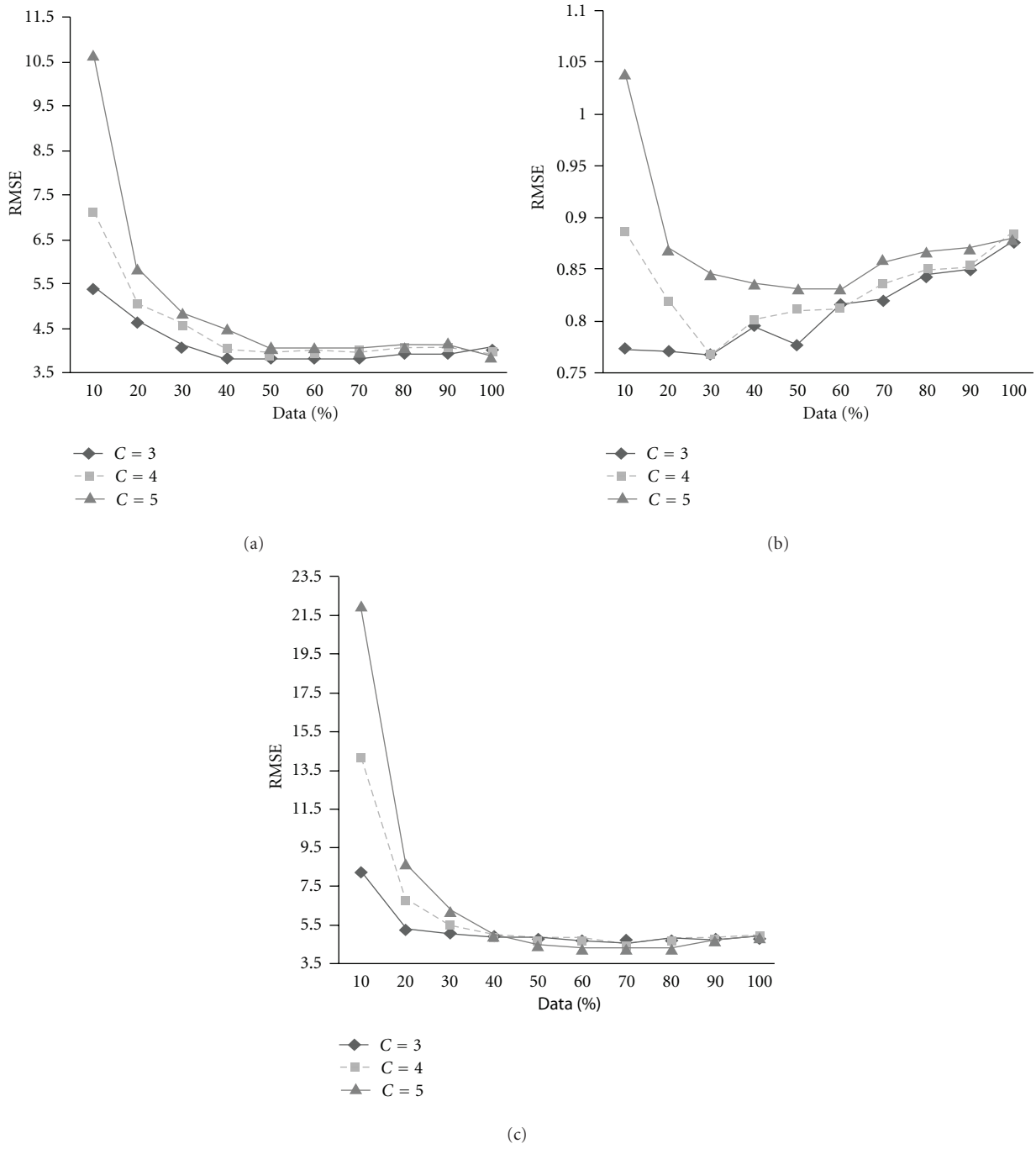

Figure 9: The values of RMSE versus the percentage of data for selected number clusters: (a) housing data, (b) PM10 data, and (c) body fat data.

data and feature reduction process. The selection of the number of cooperating swarms is important because it will affect the performance of the cooperative PSO model. Subswarm 1 represents the features' column and subswarm 2 represents the instances' row of the particular data set. Figure 4 illustrates the main difference between standard PSO and cooperative PSO. The standard PSO contains one swarm with a large dimension of search space. In contrast, for the cooperative PSO, we divide the search space into two subswarms: subswarm 1 for feature representation and subswarm 2 for instances representation. All the subswarms share the same basic particles definition illustrated in Figure 4.

In general, the dimensionality for the data (instances) selection is higher than that of the feature selection. In order to reduce the impact of the curse of dimensionality, we decompose the data into several groups by using the information granulation approach. In this paper, we used

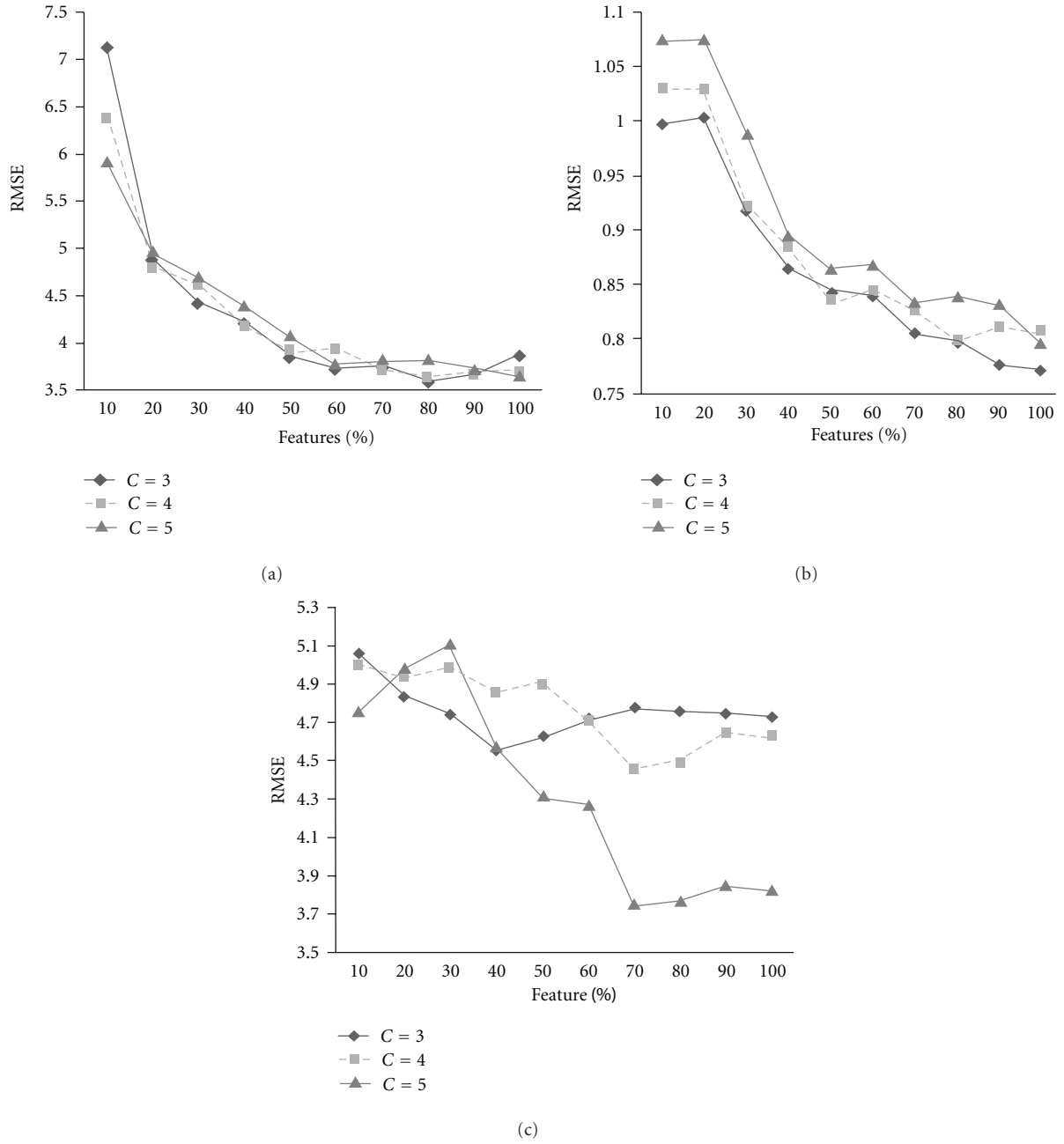

Figure 10: Plots of RMSE versus the percentage of features for selected number clusters: (a) housing data, (b) PM10 data, and (c) body Fat data.

the fuzzy C-means (FCMs) to construct the information granules. Therefore, the number of decomposition groups is actually the number of the clusters (c) used in the FCM. For example, if we want to decompose the data into three groups, we use the number of clusters equal to three. As a result, instead of having only two subswarms, we introduce more subswarms that represent different groups of data.

Figure 5 presents the process of constructing the subswarms for cooperative PSO by decomposing the instances into several subswarms. As mentioned earlier, we apply the concept of information granulation to decompose the data group. In order to identify the selected data in each decomposed group, we use the information granules (membership degrees) values to identify the index of the instances in each group. Here, we employ a winner-takes-all scheme to determine a single group for each granule, that is, the index of the instances in each of the decomposition group related to the information granule that gets the highest degrees of activation. We denote the set of data associated with the ith granules by $X(i_0)$:

$$X(i_0) = \left\{ x_k \in X \mid U_{i0k} = \max_i U_{ik} \right\} \quad \text{for } 1 \le k \le M,\ 1 \le i \le c, \tag{3}$$

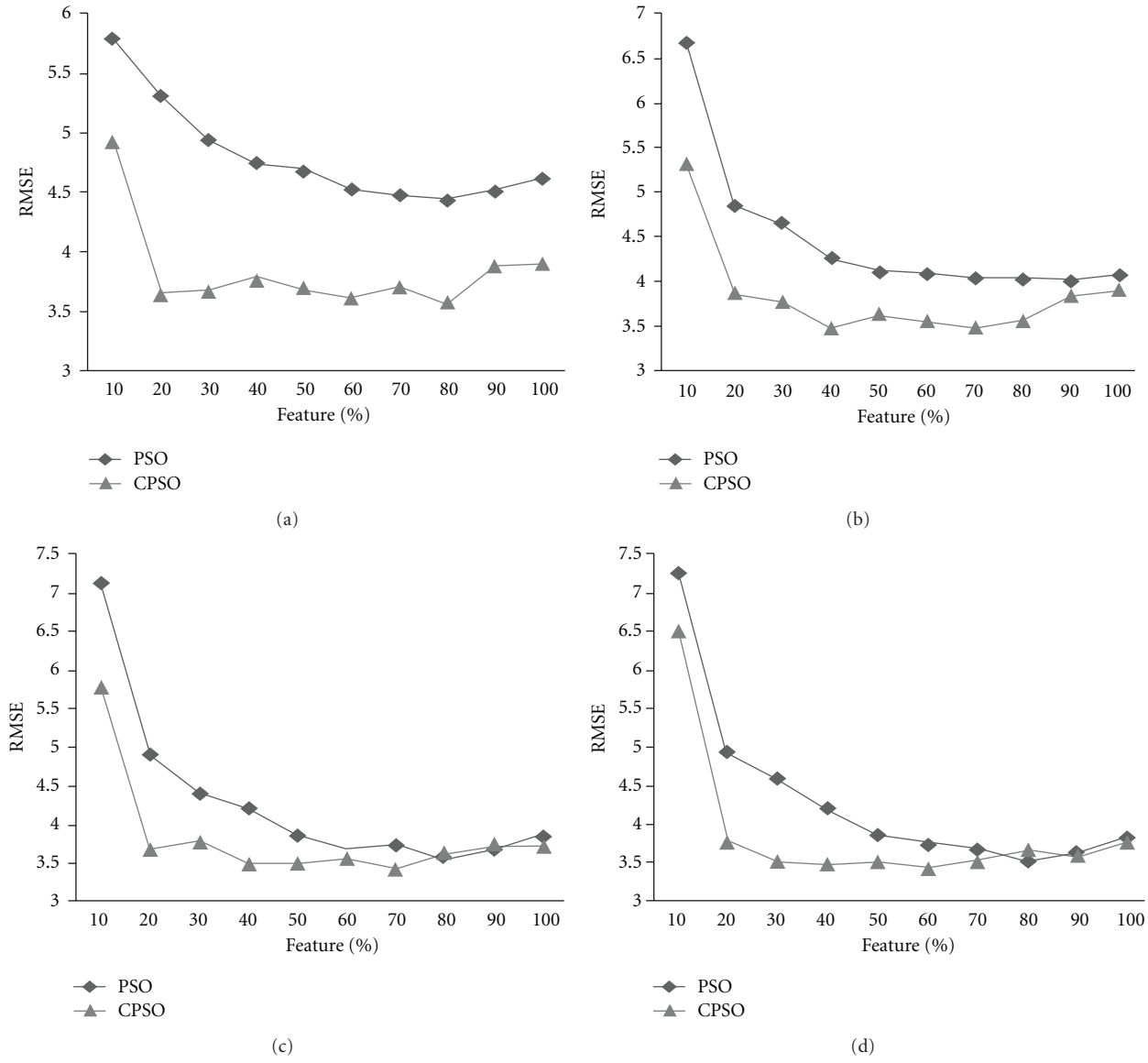

Figure 11: Values of RMSE versus the percentage of features selected when running PSO and CPSO—the use of the housing dataset: (a) 20% of selected data, (b) 30% of selected data, (c) 50% of selected data, and (d) 70% of selected data.

where $X(i_0)$ is the decomposition groups, U_{ik} is the information granules for each data, x_k is the data (instances), M and c are the number of data and the level of information granulation, respectively.

4.3. A Category of Fuzzy-Rule-Based Models. To make an overall presentation more focused, we consider a class of fuzzy-rule-based models governed by the collection of "c" rules:

$$\text{if } \mathbf{x} \text{ is } A_i \quad \text{then} \quad y = f_i(\mathbf{x}, \mathbf{a}_i), \tag{4}$$

where $i = 1, 2, \ldots, c$ (c is the number of clusters), A_i are the information granules formed in the input space, and f_i is a local linear function with some parameters $\mathbf{a}_i$ associated with the corresponding information granule. The information granules A_i are constructed by means of fuzzy clustering, namely, fuzzy C-means (FCMs). The corresponding membership functions A_i are thus described as

$$A_i(x) = \frac{1}{\sum_{j=1}^{c} \left(\|\mathbf{x} - \mathbf{v}_i\| / \left\|\mathbf{x} - \mathbf{v}_j\right\| \right)^{2(m-1)}}, \tag{5}$$

where $\mathbf{v}_i$, $i = 1, 2, \ldots, c$ are the prototypes formed through clustering, and m, $m > 1$ is a fuzzification coefficient.

5. Experimental Studies

In this section, we report our results from a set of experiments, using several machine learning data sets (see http://www.ics.uci.edu/~mlearn/MLRepository.html and http://lib.stat.cmu.edu/datasets/). The main objective of these

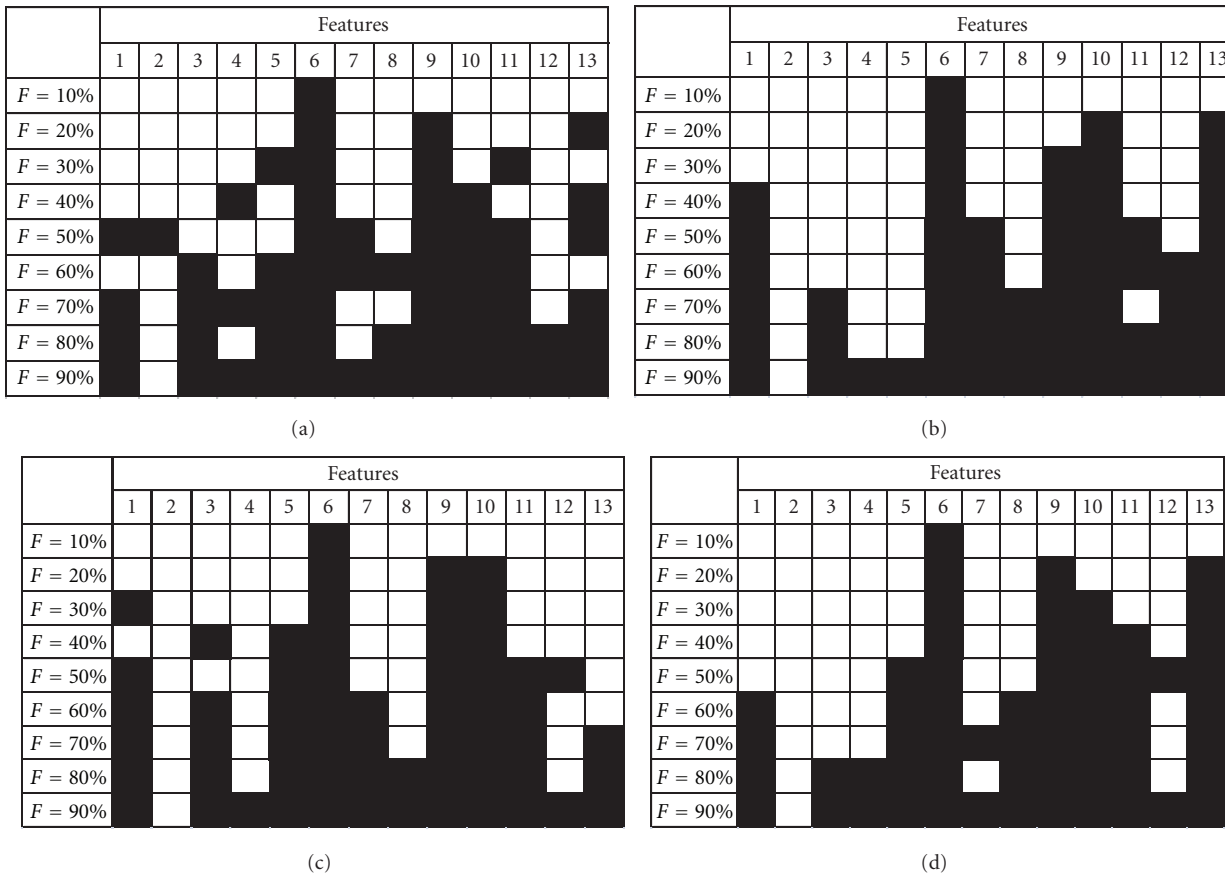

Figure 12: Comparison of sets of features being selected by using PSO and CPSO^2 for Housing data dataset: (a) PSO method with 30% of selected data, (b) CPSO^2 method with 30% of selected data, (c) PSO method with 70% of selected data, and (d) CPSO^2 method with 70% of selected data.

Table 7: The optimal % of features and data for different clusters.

Data set and number of clusters	% of features	% of data
Pima with $C = 3$	70	40
Pima with $C = 4$	100	20
Pima with $C = 5$	100	20
Pima with $C = 6$	100	40
Pima with $C = 7$	100	80
Housing with $C = 3$	80	70
Housing with $C = 4$	80	50
Housing with $C = 5$	80	100
Housing with $C = 6$	80	80
Housing with $C = 7$	90	100
Body Fat with $C = 3$	30	30
Body Fat with $C = 4$	100	70
Body Fat with $C = 5$	90	70
Body Fat with $C = 6$	90	90
Parkinson with $C = 3$	30	30

experiments is to show the abilities of the proposed approach, quantify the performance of the selected subsets of features and instances, and arrive at some general conclusions. A concise summary of the data sets used in the experiment is presented in Table 2. All the data concern continuous output.

5.1. Parameter Setup. The values of the PSO and CPSO parameters were set using the standard form as follows. The values of the inertia weight, w, were linearly from 1 to 0 over the course of optimization. The values of the cognitive factor, c_1, and social factor c_2, were set to 0.5 and 1.5, respectively. In Table 3, we also list the numeric values of the parameters of the PSO and CPSO environment. As to the size of the population and the number of generations, we used a larger population and a larger number of generations in the generic version of the PSO than in the CPSO because of the larger search space this algorithm operates in.

The number of subswarms used in the optimization method is also shown in Table 3. The PSO method comprises only a single swarm whose individuals concatenate features and instances. In contrast, for the CPSO, we divided the search space into several subswarms that can cooperate with each other and where the individuals in the subswarms are used to represent a portion of the search space. The CPSO^1 contains two subswarms that cover the data and

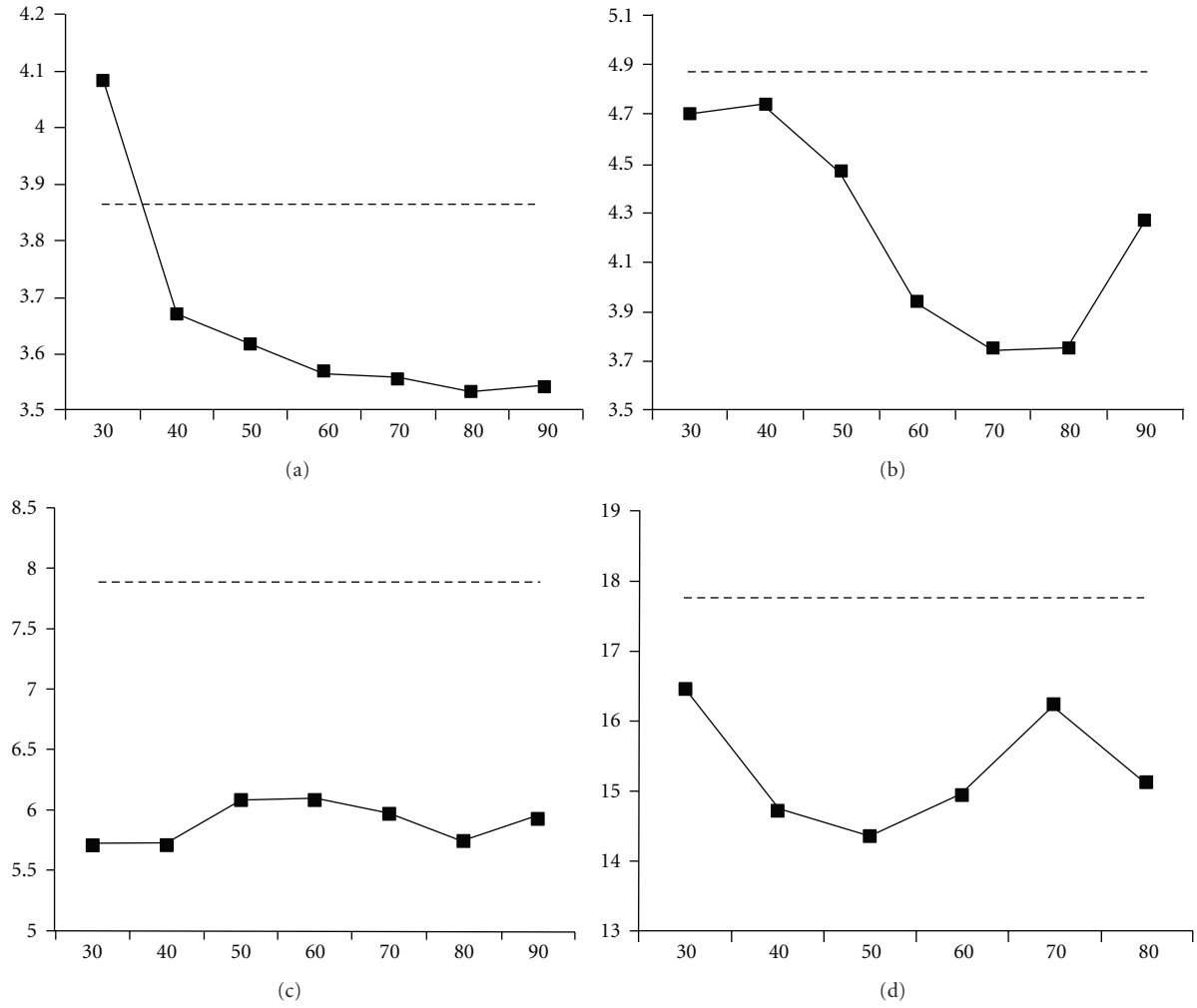

Figure 13: Comparison of RMSE by using proposed method (straight line) and standard fuzzy model (dotted line): (a) Housing dataset with $c = 4$, (b) body fat dataset with $c = 5$, (c) Parkinson's dataset with $c = 3$, and (d) computer dataset with $c = 3$.

Table 8: Best subsets of features for PM10 data.

F	Subset of features						
	$D = 30\%$	$D = 40\%$	$D = 50\%$	$D = 60\%$	$D = 70\%$	$D = 80\%$	$D = 90\%$
10%	1	1	1	1	1	1	1
20%	1	1	1	1	1	1	1
30%	1, 7	1, 7	1, 7	1, 7	1, 6	1, 6	1, 6
40%	1, 6, 7	1, 6, 7	1, 6, 7	1, 6, 7	1, 6, 7	1, 6, 7	1, 6, 7
50%	1, 2, 6, 7	1, 2, 6, 7	1, 2, 6, 7	1, 2, 6, 7	1, 2, 6, 7	1, 2, 6, 7	1, 2, 6, 7
60%	1, 2, 6, 7	1, 2, 6, 7	1, 2, 6, 7	1, 2, 6, 7	1, 2, 6, 7	1, 2, 6, 7	1, 2, 6, 7
70%	1, 2, 3, 6, 7	1, 2, 3, 6, 7	1, 2, 3, 6, 7	1, 2, 3, 6, 7	1, 2, 3, 6, 7	1, 2, 4, 6, 7	1, 2, 4, 6, 7
80%	1, 2, 3, 4, 6, 7	1, 2, 3, 5, 6, 7	1, 2, 3, 4, 6, 7	1, 2, 3, 4, 6, 7	1, 2, 3, 4, 6, 7	1, 2, 3, 4, 6, 7	1, 2, 3, 4, 6, 7
90%	1, 2, 3, 4, 6, 7	1, 2, 3, 4, 6, 7	1, 2, 3, 4, 6, 7	1, 2, 3, 5, 4, 7	1, 2, 3, 4, 5, 7	1, 2, 3, 4, 6, 7	1, 2, 3, 5, 4, 7

features, respectively. In CPSO^2, we used three subswarms to represent data point; in the data used here, the number of data is larger than the number of features, so a better balance of the dimensionality of the spaces is achieved. The data (instances) search space is divided into three subswarms, and the decomposition process is realized by running fuzzy clustering (each cluster forms a subswarm). In the table we used a smaller size of generation compared to particles size. This is because in [34] Shi and Eberhart mentioned that the population size does not exhibit any significant impact on

Table 9: Best subsets of features for body fat data.

F	Subset of features			
	$D = 50\%$	$D = 60\%$	$D = 70\%$	$D = 80\%$
10%	1	1	1	1
20%	1, 6, 7	1, 3, 7	1, 3, 7	1, 3, 7
30%	1, 6, 7, 9	1, 7, 8, 9	1, 3, 7, 9	1, 3, 7, 9
40%	1, 6, 7, 8, 9, 12	1, 3, 4, 7, 8, 9	1, 3, 6, 7, 8, 9	1, 7, 8, 9, 11, 12
50%	1, 2, 6, 7, 8, 12, 14	1, 4, 8, 9, 11, 12, 14	1, 3, 7, 8, 9, 12, 14	1, 3, 4, 5, 7, 8, 12
60%	1, 2, 6, 7, 8, 11, 12, 14	1, 3, 4, 7, 8, 9, 11, 14	1, 3, 5, 7, 9, 11, 12, 14	1, 3, 5, 7, 8, 11, 12, 14
70%	1, 3, 4, 5, 6, 8, 9, 11, 12, 14	1, 3, 4, 5, 7, 8, 9, 11, 12, 14	1, 3, 4, 5, 7, 8, 9, 11, 12, 14	1, 3, 4, 5, 7, 8, 9, 11, 12, 14
80%	1, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14	1, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14	1, 3, 4, 5, 6, 7, 8, 9, 11, 12, 14	1, 3, 4, 5, 6, 7, 8, 9, 11, 12, 14
90%	1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14	1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14	1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14	1, 23, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14

Table 10: Best subsets of features for housing data.

F	Subset of features				
	$D = 10\%$	$D = 30\%$	$D = 50\%$	$D = 70\%$	$D = 90\%$
10%	12	6	6	12	12
20%	5, 12, 13	6, 9, 13	6, 9, 10	6, 9, 10	6, 9, 10
30%	6, 7, 9, 13	5, 6, 9, 11	4, 6, 9, 13	1, 6, 9, 10	2, 3, 12, 13
40%	1, 3, 6, 10, 13	4, 6, 9, 10, 13	6, 9, 10, 12, 13	3, 5, 6, 9, 10	1, 4, 6, 9, 10
50%	3, 6, 7, 8, 9, 10, 11	1, 2, 6, 9, 10, 11, 13	1, 3, 5, 6, 8, 9, 11	1, 5, 6, 9, 10, 11, 12	1, 6, 9, 10, 11, 12, 13
60%	3, 5, 6, 7, 8, 9, 12, 13	3, 5, 6, 7, 8, 9, 10, 11	1, 3, 5, 6, 7, 9, 10, 11	1, 3, 5, 6, 7, 9, 10, 11	1, 3, 6, 7, 8, 9, 11, 13
70%	3, 5, 6, 7, 8, 9, 10, 12, 13	1, 3, 4, 5, 6, 9, 10, 11, 13	1, 3, 5, 6, 7, 9, 10, 11, 13	1, 3, 5, 6, 7, 9, 10, 11, 13	1, 3, 6, 7, 8, 9, 10, 11, 13
80%	1, 3, 4, 5, 6, 8, 9, 10, 11, 13	1, 3, 5, 6, 8, 9, 10, 11, 12, 13	1, 3, 5, 6, 7, 8, 9, 10, 11, 13	1, 3, 5, 6, 7, 8, 9, 10, 11, 13	1, 3, 5, 6, 7, 8, 9, 10, 11, 13
90%	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13	1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13	1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13	1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13	1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

Initialize m one-dimensional PSO: P_j, $j \in [1, \ldots, m]$
Create
$\mathbf{C}(j,k) = [P_{1(\mathrm{BG})}, P_{2(\mathrm{BG})}, \ldots, P_{j-1(\mathrm{BG})}, k, P_{j+1(\mathrm{BG})}, \ldots, P_{m(\mathrm{BG})}]$
While stop criteria not met **do**
  for each subswarm $j \in [1, \ldots, m]$ **do**
    for each particle $i \in [1, \ldots, s]$ **do**
      if fitness($\mathbf{C}(j, P_j(x_i))$) > fitness($\mathbf{C}(j, P_{j(\mathrm{LB}_i)})$)
        then $P_{j(\mathrm{LB}_i)} = P_j(x_i)$
      if fitness($\mathbf{C}(j, P_{j(\mathrm{LB}_i)})$) > fitness($\mathbf{C}(j, P_{j(\mathrm{BG})})$)
        then $P_{j(\mathrm{BG})} = P_{j(\mathrm{lb}_i)}$
  end for
  for each P_j do
    $v_{i,j}(t+1) = w \cdot v_{i,j}(t) + c_1 \cdot r_{1,i}(t)[P_{\mathrm{LB}_{i,j}}(t) - x_{i,j}(t)]$
      $+c_2 \cdot r_{2,i}(t)[P_{\mathrm{BG}_{i,j}}(t) - x_{i,j}(t)]$
    $x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1)$

Algorithm 1: Pseudocode for cooperative PSO.

Table 11: Standard deviations for PSO and CPSO (housing and PM10 data sets).

Housing ($D = 50\%$)		PM10 ($D = 50\%$)	
PSO	CPSO	PSO	CPSO
0.066	0.072	0.021	**0.007**
0.192	**0.015**	0.018	**0.009**
0.199	**0.039**	0.010	**0.007**
0.11	**0.093**	0.004	**0.007**
0.115	**0.079**	0.024	**0.008**
0.091	**0.071**	0.006	**0.009**
0.058	0.094	0.014	**0.010**
0.044	0.064	0.010	**0.007**
0.053	**0.042**	0.019	**0.009**
0.021	0.042	0.016	**0.009**

the performance of the PSO method. However, the size of particles is high given the size of the search space. Here we require more particles to capture the large search space of instances selection for using the standard PSO. As a result we can find the best solution faster than using a smaller particles size. On the other hand, the number of particle is decreased when we implement the CPSO method. This is because the original large search space is divided into several groups, and the processes of searching the best subset are done in parallel.

5.2. Results of the Experiments. In the experiments, we looked at the performance—an average root mean squared error (RMSE)—obtained for the selected combinations of the number of features and data (instances). The results obtained for the Housing data, PM10 data, and Parkinson's data for $c = 4$ and $c = 3$ clusters are summarized in Tables 4, 5, and 6, respectively. The experiments were repeated 10 times, and the reported results are the average RMSE values. We also report the values of the standard deviation of the performance index to offer a better insight into the variability of the performance. It is noticeable that the standard deviation is reduced with the increase of the data involved and the decrease of the dimensionality of the feature space.

The visualization of the results in the form of a series of heat maps, see Figures 6, 7, and 8, helps us arrive at a number of qualitative observations as well as to look at some quantitative relationships. In most cases, the performance index remains relatively low in some regions of the heat map. This finding demonstrates that the available data come with some evident redundancy, which exhibits a negative impact on the designed model. For the PM10 data, there is a significantly reduced performance of the model when, for a low percentage of data, the number of features starts growing. This effect is present for different numbers of clusters. The same tendency is noticeable for the other data sets. There is a sound explanation to this phenomenon: simply, the structure formed by fuzzy clustering does not fully reflect the dependencies in the data (due to the effect of the sparsity of the data), and this problem, in turn, results in the deteriorating performance of the fuzzy model. In this case, one would be better off to consider a suitable reduced set of features. In all cases experimented with, we noted an optimal combination of features and data that led to the best performance of the model. Table 7 summarizes the optimal combinations of features and data.

The relationships between the percentage of data used and the resulting RMSE values are displayed in Figures 9 and 10. Some interesting tendencies are worth noting. A critical number of data are required to form a fuzzy model. Increasing the number of data does not produce any improvement as the curves plotted on Figures 9(a), 9(c), and 9(a) achieve a plateau or even some increase of the RMSE is noticeable.

Considering a fixed percentage of the data used, we look at the nature of the feature sets. Tables 8, 9 and 10 displays the best feature for PM10 data, Body fat data, and Housing data, respectively. Overall, the selected subsets of features are almost the same for different numbers of the clusters being used. Furthermore, we observe that in most cases, the reduced feature spaces exhibit an interesting "nesting" property, meaning that the extended feature space constructed subsumes the one formed previously. For example, for the Housing data, we obtain the following subsets of features:

$$\begin{aligned}&\{\text{feature 6}\} \subset \{\text{feature 6, feature 9, feature 13}\}\\ &\subset \{\text{feature 6, feature 9, feature 10, feature 13}\}.\end{aligned} \tag{6}$$

Here, the corresponding features are as follows: 6: average number of rooms per dwelling, 9: index of accessibility to radial highways, 13: percentage of lower status population, and 10: full-value property-tax rate per \$10,000. This combination is quite convincing.

For the PM10 data, we arrive at a series of nested collections of features:

$$\begin{aligned}&\{\text{feature 1}\} \subset \{\text{feature 1, feature 7}\}\\ &\subset \{\text{feature 1, feature 6, feature 7}\}\\ &\subset \{\text{feature 1, feature 2, feature 6, feature 7}\},\end{aligned} \tag{7}$$

where the corresponding features include: 1: the concentration of PM10 (particles), 7: hour of experiment per day, 6: wind direction, and 2: the number of cars per hour.

Turning to the comparative analysis of performance of the swarm optimization methods, we summarize the obtained results in Figure 11. For all data, the CPSO performed better than the standard PSO. Although both algorithms show the same tendency when the percentage of feature is 100% however, the RMSE produced by the CPSO is lower than the one obtained when running the PSO. Furthermore, the CPSO algorithm is more stable than the standard PSO. In most cases, the standard deviations of error produced by the CPSO are smaller than the results obtained for the standard PSO (see Table 11).

Figure 12 shows the subsets of the features selected for different percentages of the features used in construction of the fuzzy model. The CPSO algorithm is more consistent while selecting the increasing number of features. For

Table 12: Percentage of improvement of the RMSE obtained when using CPSO over the results formed by the PSO; Housing data set.

F	$D = 10\%$	$D = 20\%$	$D = 30\%$	$D = 40\%$	$D = 50\%$	$D = 60\%$	$D = 70\%$	$D = 80\%$	$D = 90\%$
10%	13	15	20	26	19	18	10	19	11
20%	31	31	20	24	25	16	24	23	17
30%	33	26	19	20	14	15	24	18	21
40%	28	21	19	14	17	17	17	18	19
50%	34	21	12	7	9	9	9	12	8
60%	23	21	13	7	4	7	8	9	7
70%	19	17	14	4	9	6	4	8	8
80%	33	19	12	4	2	3	3	8	9
90%	22	14	4	5	2	1	2	2	1
100%	17	4	4	7	3	1	1	1	1

Table 13: The comparison of RMSE obtained when using standard PSO, CPSO, and standard fuzzy model with holdout method for housing data with $C = 3$.

% of data	Standard PSO		Cooperative PSO[1]		Holdout method	
	% of feature	MSE	% of feature	MSE	% of feature	MSE
30	90	4.015	40	3.473	100	17.593
40	80	3.699	70	3.464	100	10.803
50	80	3.573	70	3.414	100	9.907
60	80	3.556	70	3.435	100	8.507
70	80	3.527	60	3.413	100	8.312
80	80	3.654	90	3.449	100	8.164
90	80	3.679	90	3.615	100	7.641

Table 14: The comparison of RMSE obtained when using standard PSO, CPSO, and standard fuzzy model with holdout method for body fat data with $C = 3$.

% of data	Standard PSO		Cooperative PSO[1]		Holdout method	
	% of feature	MSE	% of feature	MSE	% of feature	MSE
30	30	4.677	30	4.6847	100	11.586
40	30	4.717	30	4.5409	100	8.291
50	40	4.617	30	4.5136	100	7.548
60	50	4.636	40	4.4289	100	7.073
70	40	4.548	40	4.4234	100	6.658
80	40	4.553	40	4.4233	100	6.239
90	40	4.582	40	4.3771	100	6.102

Table 15: The comparison of RMSE obtained when using standard PSO, CPSO, and standard fuzzy model with holdout method for PM10 data with $C = 3$.

% of data	Standard PSO		Cooperative PSO[1]		Holdout method	
	% of feature	MSE	% of feature	MSE	% of feature	MSE
30	100	0.764	80	0.7338	100	2.100
40	100	0.765	80	0.7432	100	2.018
50	90	0.777	90	0.7790	100	2.030
60	80	0.805	80	0.7769	100	1.986
70	90	0.820	80	0.8052	100	2.001
80	80	0.839	90	0.8206	100	1.983
90	70	0.847	90	0.8417	100	1.976

Table 16: The comparison of RMSE obtained when using CPSO and standard fuzzy model with holdout method for computer data with $C = 3$.

% of data	Cooperative PSO		Holdout method	
	% of feature	MSE	% of feature	MSE
30	40	16.446	100	17.453
40	30	14.712	100	17.524
50	30	14.350	100	17.680
60	40	14.935	100	17.837
70	40	16.237	100	17.918
80	40	15.122	100	18.351

example, features 6 and 13 were selected when using both 30% and 70% of data. In contrast to the selection made with the PSO algorithm, the subset of the features selected here is not as stable, especially when using only 30% of data.

Table 12 presents the percentage of the improvement when using the CPSO algorithm compared to the PSO algorithm. Note that in this percentage we included all different combinations of the features' percentages and the data percentages being used. The percentage of the improvement is higher when dealing with a smaller percentage of features and data. For example, the percentage of improvement is 34% for 10% of the instances and 50% of the features selected while the percentage of improvement is less than 10% for 60% of instances and features used. These results occurred because the PSO method has to deal with a large search space for selecting a small subset of features and instances. In contrast to the search space for CPSO, the large search space is decomposed into multiple subswarms that reduce the dimensionality of the original search space.

Tables 13, 14, 15, and 16 show the comparison of RMSE when using the proposed method and the standard fuzzy modeling method. Here the standard fuzzy model is constructed without using any feature and instances selection and the holdout method is used to select the data based on the percentage given. The experiment for using the standard fuzzy modeling is repeated for 50 times. If we analyze the tables, we can observe that our proposed method outperforms the standard method of constructing the fuzzy model from the dataset. This can be seen clearly when using the CPSO method to search for the best subset of feature and instances. For example, in Table 13 if we use the CPSO method, the RMSE for using 70% of data is 3.413, whereas the RMSE for the standard method is 8.312. The same tendency occurs for all datasets used here.

Figure 13 shows the comparison plot between the proposed method and the "standard" fuzzy modeling. In most of the cases, the proposed method showed better performance.

It becomes clear that one is able to reduce the input data in terms of the number features and instances. Moreover, the flexibility of choosing the reduction level helps the user focus on the most essential subsets of data and features (variables). The knowledge acquired about the best subset of data can be used for future data collection. In addition, the user can put more effort analyzing only the best subset of data that give more impact to the overall prediction.

6. Conclusions

In this paper, we proposed a simple framework for constructing fuzzy modeling from high-dimensional and large data. This framework has several advantages that make it better suited than other frameworks for sharing various real-life problems. Firstly, the simultaneous feature and instances selection is easily adapted to construct the structure of the fuzzy model. Secondly, the best selected subset of data obtained with this framework is capable of representing the original large data set. Thirdly, we construct an optimal (or suboptimal) collection of features and data based on the PSO. In addition, a cooperative PSO is developed in order to overcome the limitation of using standard PSO when dealing with a high-dimensional search space. The size of the selected features and data used to construct the fuzzy model can be adjusted based upon the feedback provided in terms of the performance of the model constructed for the currently accepted.

The effectiveness of the framework was validated by using four well-known regression data sets. The experiment results showed that the proposed fuzzy modeling framework is able to handle high dimensionality and a large data set simultaneously. Moreover, the curse of dimensionality problem in fuzzy modeling was substantially reduced.

In the future work one could concentrate on improving the cooperative PSO by fine-tuning the parameters of the method such as, for example, the cognitive and social parameter.

Acknowledgments

Support from the Ministry of Higher Education (MOHE) Malaysia and Universiti Teknikal Malaysia Melaka (UTeM) is gratefully acknowledged.

References

[1] W. Pedrycz and F. Gomide, *Fuzzy Systems Engineering*, John Wiley & Sons, Hoboken, NJ, USA, 2007.

[2] G. Castellano, C. Castiello, A. M. Fanelli, and C. Mencar, "Knowledge discovery by a neuro-fuzzy modeling framework," *Fuzzy Sets and Systems*, vol. 149, no. 1, pp. 187–207, 2005.

[3] Y. Jin, "Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 2, pp. 212–221, 2000.

[4] Q. Zhang and M. Mahfouf, "A hierarchical Mamdani-type fuzzy modelling approach with new training data selection and multi-objective optimisation mechanisms: a special application for the prediction of mechanical properties of alloy steels," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2419–2443, 2011.

[5] G. E. Tsekouras, "On the use of the weighted fuzzy c-means in fuzzy modeling," *Advances in Engineering Software*, vol. 36, no. 5, pp. 287–300, 2005.

[6] A. G. Di Nuovo, M. Palesi, and V. Catania, "Multi-objective evolutionary fuzzy clustering for high-dimensional problems," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1–6, July 2007.

[7] M. Setnes, R. Babuška, U. Kaymak, and H. R. Van Nauta Lemke, "Similarity measures in fuzzy rule base simplification," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 28, no. 3, pp. 376–386, 1998.

[8] M. Y. Chen and D. A. Linkens, "Rule-base self-generation and simplification for data-driven fuzzy models," *Fuzzy Sets and Systems*, vol. 142, no. 2, pp. 243–265, 2004.

[9] H. Wang, S. Kwong, Y. Jin, W. Wei, and K. F. Man, "Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction," *Fuzzy Sets and Systems*, vol. 149, no. 1, pp. 149–186, 2005.

[10] F. J. Berlanga, A. J. Rivera, M. J. del Jesus, and F. Herrera, "GP-COACH: genetic Programming-based learning of COmpact and ACcurate fuzzy rule-based classification systems for High-dimensional problems," *Information Sciences*, vol. 180, no. 8, pp. 1183–1200, 2010.

[11] J. Alcalá-Fdéz, R. Alcalá, and F. Herrera, "A fuzzy associative classification system with genetic rule selection for high-dimensional problems," in *Proceedings of the 4th International Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS '10)*, pp. 33–38, March 2010.

[12] Y. Chen, B. Yang, A. Abraham, and L. Peng, "Automatic design of hierarchical Takagi-Sugeno type fuzzy systems using evolutionary algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 3, pp. 385–397, 2007.

[13] M. R. Delgado, F. V. Zuben, and F. Gomide, "Coevolutionary genetic fuzzy systems: a hierarchical collaborative approach," *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 89–106, 2004.

[14] N. Xiong and L. Litz, "Reduction of fuzzy control rules by means of premise learning—method and case study," *Fuzzy Sets and Systems*, vol. 132, no. 2, pp. 217–231, 2002.

[15] A. E. Gaweda, J. M. Zurada, and R. Setiono, "Input selection in data-driven fuzzy modeling," in *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, vol. 3, pp. 1251–1254, December 2001.

[16] M. L. Hadjili and V. Wertz, "Takagi-Sugeno fuzzy modeling incorporating input variables selection," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 6, pp. 728–742, 2002.

[17] R. Šindelář and R. Babuška, "Input selection for nonlinear regression models," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 5, pp. 688–696, 2004.

[18] M. H. F. Zarandi, I. B. Türkşen, and B. Rezaee, "A systematic approach to fuzzy modeling for rule generation from numerical data," in *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS '04)*, pp. 768–773, June 2004.

[19] H. Du and N. Zhang, "Application of evolving Takagi-Sugeno fuzzy model to nonlinear system identification," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 676–686, 2008.

[20] S. N. Ghazavi and T. W. Liao, "Medical data mining by fuzzy modeling with selected features," *Artificial Intelligence in Medicine*, vol. 43, no. 3, pp. 195–206, 2008.

[21] Y. Zhang, X. B. Wu, Z. Y. Xing, and W. L. Hu, "On generating interpretable and precise fuzzy systems based on Pareto multi-objective cooperative co-evolutionary algorithm," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 1284–1294, 2011.

[22] F. Wan, H. Shang, L. X. Wang, and Y. X. Sun, "How to determine the minimum number of fuzzy rules to achieve given accuracy: a computational geometric approach to SISO case," *Fuzzy Sets and Systems*, vol. 150, no. 2, pp. 199–209, 2005.

[23] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[24] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[25] H. Liu and H. Motoda, *Computational Methods of Feature Selection*, Chapman & Hall/CRC, Boca Raton, Fla USA, 2008.

[26] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, and J. Kittler, "A review of instance selection methods," *Artificial Intelligence Review*, vol. 34, no. 2, pp. 133–143, 2010.

[27] H. Liu and H. Motoda, *Instance Selection and Construction for Data Mining*, Kluwer Academic Publishers, Boston, Mass USA, 2001.

[28] H. Ishibuchi, T. Nakashima, and M. Nii, "Genetic-Algorithm-Based instance and feature selection," in *Instance Selection and Construction for Data Mining*, H. Lui and H. Motoda, Eds., pp. 95–112, Kluwer Academic Publishers, Boston, Mass, USA, 2001.

[29] J. Derrac, S. García, and F. Herrera, "IFS-CoCo: instance and feature selection based on cooperative coevolution with nearest neighbor rule," *Pattern Recognition*, vol. 43, no. 6, pp. 2082–2105, 2010.

[30] A. Hertz and D. Kobler, "Framework for the description of evolutionary algorithms," *European Journal of Operational Research*, vol. 126, no. 1, pp. 1–12, 2000.

[31] J. R. Cano, F. Herrera, and M. Lozano, "Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 6, pp. 561–575, 2003.

[32] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, December 1995.

[33] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[34] Y. Shi and R. C. Eberhart, "Empirical study of particle Swarm Optimization," *Congress on Evolutionary Computing*, vol. 3, pp. 1945–1950, 1999.