Intuitionistic Neuro-Fuzzy Network with Evolutionary Adaptation

Petr Hájek[1], Vladimír Olej[1]

[1]Institute of System Engineering and Informatics, Faculty of Economics and Administration, University of Pardubice, Pardubice, Czech Republic
Petr.Hajek {Vladimir.Olej}@upce.cz

Abstract. Intuitionistic fuzzy inference systems (IFISs) incorporate imprecision in the construction of membership functions present in fuzzy inference systems. In this paper we design intuitionistic neuro-fuzzy networks to adapt the antecedent and consequent parameters of IFISs. We also propose a mean of maximum defuzzification method for a class of Takagi-Sugeno IFISs and this method is compared with the basic defuzzification distribution operator. On both real-life credit scoring data and seven benchmark regression datasets we show that the intuitionistic neuro-fuzzy network trained with particle swarm optimization outperforms traditional ANFIS methods (hybrid and backpropagation) and ANFIS trained with evolutionary algorithms (genetic algorithm and particle swarm optimization), respectively. A set of nonparametric tests for multiple datasets is performed to demonstrate statistical differences between the algorithms. In the task of adapting the intuitionistic neuro-fuzzy network, we show that particle swarm optimization provides a higher prediction accuracy compared with traditional algorithms based on gradient descent or least-squares estimation.

Keywords: ANFIS, intuitionistic fuzzy sets, intuitionistic fuzzy inference systems of Takagi-Sugeno type, intuitionistic neuro-fuzzy network, defuzzification method, particle swarm optimization.

1 Introduction

Fuzzy inference systems (FISs) are one of the most widely used tools to model highly nonlinear systems with uncertainty. By incorporating imprecision into the models, FISs have provided good generalization performance in various applications. In addition, using linguistic terms makes the models descriptive and interpretable in their respective domains. However, determining the precise membership functions (both in antecedents and consequents of if-then rules) is problematic due to uncertainties associated with linguistic terms, disagreement among experts, or noise in the data (Liang and Mendel 2000; Hagras and Wagner 2012). Therefore, researchers and practitioners call for additional freedom in the design of membership functions, making it possible to minimize the effects of the mentioned uncertainties (Zarandi et al. 2009). As a result, two main categories of FIS generalizations have been proposed, interval-valued FISs (Mendel 2006) and intuitionistic FISs (Olej and Hájek 2010; Olej and Hájek 2011; Hájek and

Olej 2012; Hájek and Olej 2014). Currently, algorithms are being developed to optimize additional uncertainty in FISs' parameters.

Neuro-fuzzy networks have shown the ability to model non-linear systems at high accuracy, representing an implementation of FIS to adaptive networks for developing fuzzy if-then rules with suitable membership functions (Kasabov 2015; Demertzis et al. 2016). For example, ANFIS (Shing and Jang 1993; Loganathan and Girija 2013) identifies a set of parameters through several learning rules: gradient descent algorithm, hybrid algorithm (combining back-propagation gradient descent and least-squares method), and Kalman filter.

The concept of intuitionistic fuzzy sets (IF-sets) (Atanassov 1986; Atanassov 1999) was developed as an alternative approach to define a fuzzy set in cases where available information is not sufficient for the definition of an imprecise concept by means of a conventional fuzzy set, but the concept can be more naturally approached by separately envisaging positive and negative instances (Dubois and Prade 2005). As a result, loosely related membership and non-membership functions can be defined in intuitionistic FISs (IFISs). This approach has been preferred in various application domains such as control (Akram et al. 2014), time series analysis (Castilo et al. 2007), air pollution prediction (Olej and Hájek 2010), or bankruptcy forecasting (Hájek and Olej 2014). However, these applications raised several questions regarding the adaptation of IFISs, most noticeably (1) which algorithms are effective for setting and fine-tuning the parameters of the IFISs and (2) how to defuzzify the IF-sets. This paper seeks to address the questions in the following way: (1) an intuitionistic neuro-fuzzy network (INFN) with evolutionary adaptation is developed to optimize the antecedent and consequent parameters of the IFIS, and (2) the mean of maximum (MOM) defuzzification method is proposed for a class of Takagi-Sugeno IFISs. The antecedent and consequent parameters of the IFIS are adapted using particle swarm optimization (PSO) as evolutionary algorithms have shown promising results in FIS adaptation (Angelov 2012; Maciel et al. 2012; Henzgen et al. 2014).

As an application, we first investigate whether INFNs with evolutionary adaptation can be effectively employed in credit scoring. Credit scoring has become an important application domain of soft computing because it is based on a group of expert decision makers and, moreover, the determinants of credit scoring and their weight are associated with a high degree of uncertainty (Hájek 2012). In the second set of experiments, we investigate the effectiveness of the INFNs with evolutionary adaptation on several benchmark regression datasets. We also perform statistical analyses of the results to show the dominance of the proposed method over the ANFIS trained with traditional algorithms.

The remainder of this paper has been organized in the following way. Section 2 presents the design of INFN which is based on the IFIS of a first-order Takagi-Sugeno type. Section 3 provides an overview of defuzzification methods for this class of systems. Section 4 presents methods for adapting the parameters of INFNs. Section 5 describes the credit scoring dataset and the preprocessing of input textual data. In this section, we also briefly introduce the benchmark datasets used for the comparative statistical analysis of prediction performance. Section 6 presents the results of experiments in terms of RMSE, comparing the performance of the INFN and ANFIS adapted with various learning algorithms and defuzzification methods. Section 7 concludes the paper and discusses possible future research directions.

2 Intuitionistic Neuro-Fuzzy Network

Let a set X be a non-empty set. An IF-set A in X is an object having the form $A=\{\langle x,\mu_A(x),\nu_A(x)\rangle|x\in X\}$, where the function $\mu_A:X\rightarrow[0,1]$ defines the degree of membership function $\mu_A(x)$ and the function $\nu_A:X\rightarrow[0,1]$ defines the degree of non-membership function $\nu_A(x)$, respectively, of the element $x\in X$ to the set A, $A\subset X$. For every $x\in X$, $0\leq\mu_A(x)+\nu_A(x)\leq 1$, $\forall x\in X$ must hold (Atanassov 1986; Atanassov 1999). The amount $\pi_A(x)=1-(\mu_A(x)+\nu_A(x))$ is called the hesitation part (intuitionistic index, IF-index) and represents a measure of non-determinacy.

In the IFIS of a first-order Takagi-Sugeno type (Olej and Hájek 2011), the k-th if-then rule R_k, k=1,2, ... ,N, is defined as follows

R_k: **if** x_1 is $A_{1,k}$ AND x_2 is $A_{2,k}$ AND ... AND x_i is $A_{i,k}$ AND ... AND x_n is $A_{n,k}$

then $y_k=f(x_1,x_2, ... ,x_n)=a_{0,k}+a_{1,k}x_1+a_{2,k}x_2+ ... +a_{i,k}x_i+ ... +a_{n,k}x_n$, (1)

where $A_{1,k},A_{2,k}, ... ,A_{i,k}, ... ,A_{n,k}$ represent IF-sets and y_k is the output of the k-th rule represented by a linear combination of inputs $x_1,x_2, ... ,x_n$. The firing weight of each if-then rule is obtained using the application of t-norm operators, see (Hájek and Olej 2012) for an overview. The final defuzzified output is usually calculated as the weighted average of each if-then rule's output.

The design of INFN is based on the ANFIS model. Typically, there exist six layers in this model (Fig. I).

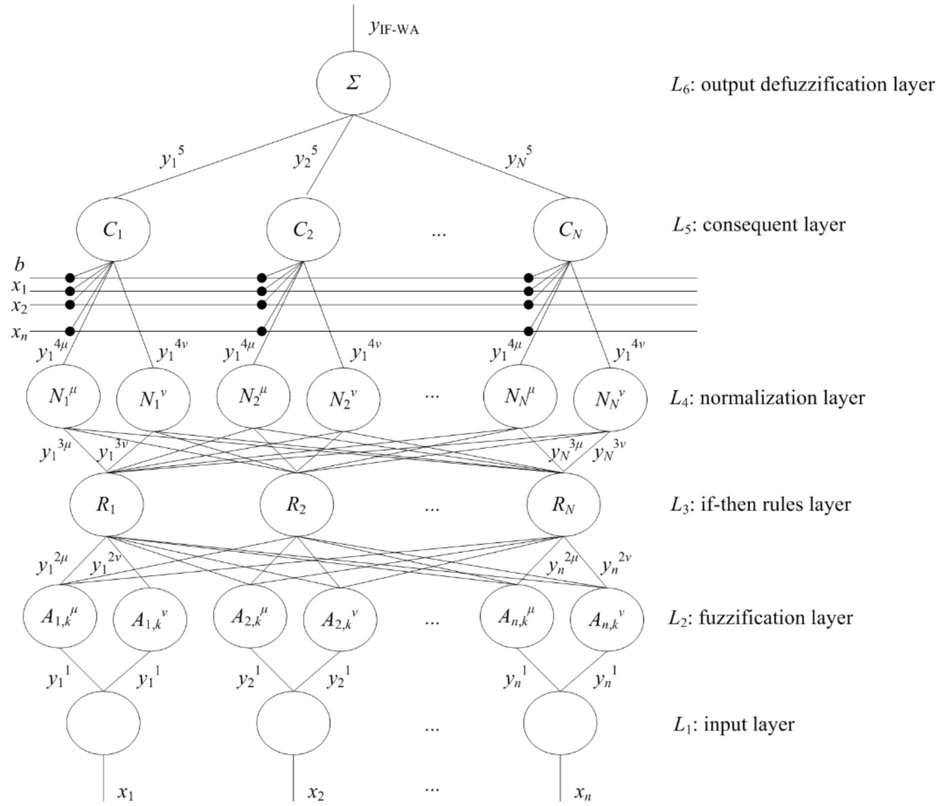

Fig. I INFN model

Layer 1: This first layer is the input layer of the INFN. Neurons in this layer transmit the external crisp input to the next layer. Namely, $x_1, x_2, \dots, x_n$ are the inputs and $y^1{}_1, y^1{}_2, \dots, y^1{}_n$ are the outputs of the neurons in the first layer.

Layer 2: Neurons in the second layer represent antecedent IF-sets of if-then rules. Here, a fuzzification neuron receives an input and determines the degree to which this signal belongs to the neuron's IF-set. In this case, linguistic variables $A^{\mu}{}_{1,k}, A^{\mu}{}_{2,k}, \dots, A^{\mu}{}_{i,k}, \dots, A^{\mu}{}_{n,k}$ and $A^{\nu}{}_{1,k}, A^{\nu}{}_{2,k}, \dots, A^{\nu}{}_{i,k}, \dots, A^{\nu}{}_{n,k}$ determine membership functions $\mu(x_1), \mu(x_2), \dots, \mu(x_n)$ and non-membership functions $\nu(x_1), \nu(x_2), \dots, \nu(x_n)$), respectively, where for example $\mu(x_1)=1-\nu(x_1)-\pi(x_1)$, and $\nu(x_1)=1-\mu(x_1)-\pi(x_1)$.

If we have x_i^2 be the input and y_i^2 be the output signal of neuron i in the second layer, then we have $y_i^{2\mu}= f_2(x_i^{2\mu})$, $y_i^{2\nu}= f_2(x_i^{2\nu})$, where f_2 represents the activation function of neuron i, and is set to certain membership function $\mu(x)$ and non-membership function $\nu(x)$, usually Gaussian, bell, triangular, or trapezoidal. For example, Gaussian membership and non-membership functions can be defined as follows

$$\mu(x)=(1-\pi_a(x))\mathrm{e}^{-\alpha},\ \nu(x)=(1-\pi_b(x))-\mu(x)=1-\pi_b(x)-(1-\pi_a(x))\mathrm{e}^{-\alpha}, \qquad (2)$$

where $\alpha = (x-b)^2/2\sigma^2$ and $\pi_a(x)$ is the IF-index of center, and $\pi_b(x)$ is the IF-index of variance. Parameters b, σ, $\pi_a(x)$, and $\pi_b(x)$ are premise parameters.

Layer 3: Each neuron in the third layer corresponds to an if-then rule of the first-order Takagi-Sugeno type IFIS. An if-then rule neuron receives signals from the fuzzification neurons (involved in the antecedents of the if-then rule) and computes the firing weight of the if-then rule in the following way. Let $w^{\mu}{}_k$ and $w^{\nu}{}_k$ be firing weights which are computed using Gödel t-norm (Klement et al. 2004; Deschrijver et al. 2004; Barrenchea 2009)

$$w^{\mu}{}_k = \underset{k=1,2,\ldots,N}{\mathrm{MIN}} \left(\mu(x_1),\mu(x_2), \ldots ,\mu(x_n)\right),\ w^{\nu}{}_k = \underset{k=1,2,\ldots,N}{\mathrm{MAX}} \left(\nu(x_1),\nu(x_2), \ldots ,\nu(x_n)\right). \tag{3}$$

Then, if we have x_i^3 be the input and y_i^3 be the output signal of neuron i in the third layer, then we have $y_i^{3\mu}=f_3(x_i^{3\mu})$, $y_i^{3\nu}= f_3(x_i^{3\nu})$, where f_3 represents the MAX (MIN) operators of neuron i, and we obtain $y_i^{3\mu}=(w_1^{\mu},w_2^{\mu}, \ldots ,w_N^{\mu})$ and $y_i^{3\nu}=(w_1^{\nu},w_2^{\nu}, \ldots ,w_N^{\nu})$.

Layer 4: Neuron i in the fourth layer calculates the ratio of the i-th firing weight to the sum of all rules' firing weights. For convenience, the outputs of this layer are so-called normalized firing weights of the corresponding if-then rule.

If we have x_i^4 be the input and y_i^4 be the output signal of neuron i in the fourth layer, then we have $y_i^{4\mu}=f_4(x_i^{4\mu})$ and $y_i^{4\nu}= f_4(x_i^{4\nu})$, where f_4 represents the normalized function of neuron i. Then, $y_i^{4\mu}=(\overline{w}_1^{\,\mu}, \overline{w}_2^{\,\mu}, \ldots , \overline{w}_N^{\,\mu})$ and $y_i^{4\nu}= (\overline{w}_1^{\,\nu}, \overline{w}_2^{\,\nu}, \ldots , \overline{w}_N^{\,\nu})$ are normalized values.

Layer 5: The fifth layer, the fourth hidden layer, represents consequent parameters. Each neuron in this layer is connected to the if-then rule neurons in the fourth layer. A neuron in the fifth layer computes the weighted consequent value of a given rule in the following way

$$y_i^5=x_i^5(x_1,x_2, \ldots ,x_n)= x_i^5(a_{0,k} + a_{1,k}x_1 + a_{2,k}x_2 + \ldots + a_{i,k}x_i + \ldots + a_{n,k}x_n), \tag{4}$$

where x_i^5 is the input and y_i^5 is the output signal of neuron i in the fourth layer and $a_{0,k},a_{1,k},a_{2,k}, \ldots ,a_{n,k}$, is a set of consequent parameters.

Layer 6: The sixth layer is both the output layer and the defuzzification layer. There is only one neuron in the layer, which calculates the weighted average of outputs of all neurons in the fifth layer and consequently produces the defuzzified output y_{IF_WA}.

3 Defuzzification Methods in Intuitionistic Neuro-Fuzzy Network

In related literature, several defuzzification methods have been proposed for Mamdani type IFIS (Angelov 1995; Angelov 2001; Hájek and Olej 2014) such as center of area (IF_COA), basic defuzzification distribution operator (IF_BADD), and mean of maximum (IF_MOM). IF_COA and IF_BADD have recently been adjusted to Takagi-

Sugeno type IFISs (Hájek and Olej 2014). Here, we present an overview of the defuzzification methods and, in addition, we propose the IF_MOM method based on the weighted average and weighted sum defined by Angelov (1995).

Let w^{μ}_k and w^{ν}_k be firing weights which are computed using Gödel t-norm (Klement et al. 2004; Deschrijver 2004; Barrenchea 2009). Then, when $w_k = w^{\mu}_k - w^{\nu}_k$, the weighted average IFWA1 and the weighted sum IFWS1 can be expressed as

$$y_{\mathrm{IFWA1}} = \frac{\sum_{k=1}^{N} y_k \left(w_k^{\mu} - w_k^{\nu}\right)}{\sum_{k=1}^{N} \left(w_k^{\mu} - w_k^{\nu}\right)},\ y_{\mathrm{IFWS1}} = \sum_{k=1}^{N} y_k \left(w_k^{\mu} - w_k^{\nu}\right), \text{ for } w_k^{\mu} - w_k^{\nu} > 0. \quad (5)$$

For these operators it holds that $y_{\mathrm{IFWA1}} = y^{\mu}_{\mathrm{WA1}}$ and $y_{\mathrm{IFWS1}} = y^{\mu}_{\mathrm{WS1}}$ for $w^{\nu}_k = 0$. In that case, they can be defined as traditional weighted averages (WAs) or weighted sums (WSs) of all N if-then rules R_k as follows

$$y^{\mu}_{\mathrm{WA1}} = \frac{\sum_{k=1}^{N} y_k^{\mu} w_k^{\mu}}{\sum_{k=1}^{N} w_k^{\mu}},\ y^{\mu}_{\mathrm{WS1}} = \sum_{k=1}^{N} y_k^{\mu} w_k^{\mu}. \quad (6)$$

The defuzzification methods IFWA1 and IFWS1 are defined based on the difference between the firing weights w^{μ}_k and w^{ν}_k, which have to be positive. The methods are analogical to the defuzzification method IF_COA proposed by Angelov (1995), as they provide all possible solutions in which the degree of acceptance is higher than the degree of non-acceptance $w^{\mu}_k > w^{\nu}_k$. Analogically to IF_COA, they may average good and poor solutions since only the difference between the firing weights w^{μ}_k and w^{ν}_k matter.

Further, IFWA2 and IFWS2 are defined by analogy to IF_BADD (Angelov 1995) as follows

$$y_{\mathrm{IFWA2}} = \frac{\sum_{k=1}^{N} y_k \left(w_k^{\mu} - w_k^{\nu}\right)^{\alpha}}{\sum_{\mathrm{k}=1}^{N} \left(w_k^{\mu} - w_k^{\nu}\right)^{\alpha}},\ y_{\mathrm{IFWS2}} = \sum_{k=1}^{N} y_k \left(w_k^{\mu} - w_k^{\nu}\right)^{\alpha}, \text{ for } w_k^{\mu} - w_k^{\nu} > 0. \quad (7)$$

For these operators it holds that $y_{\mathrm{IFWA2}} = y^{\mu}_{\mathrm{WA1}}$ and $y_{\mathrm{IFWS2}} = y^{\mu}_{\mathrm{WS1}}$ for $w^{\nu}_k = 0$, and $y_{\mathrm{IFWA2}} = y_{\mathrm{IFWA1}}$ and $y_{\mathrm{IFWS2}} = y_{\mathrm{IFWS1}}$ for $\alpha = 1$. For $\alpha = 0$, the defuzzified outputs y_{IFWA2} and y_{IFWS2} can be expressed in the following way

$$y_{\mathrm{IFWA2}} = \frac{\sum_{k=1}^{N} y_k}{N},\ y_{\mathrm{IFWS2}} = \sum_{k=1}^{N} y_k\ . \quad (8)$$

Alternatively, the defuzzified outputs can be defined as the WA (WS) of if-then rules R_k maximizing the difference between w^{μ}_k and w^{ν}_k. Thus, the defuzzification methods can be defined by analogy to IF_MOM method proposed by Angelov (1995) as follows

$$y_{\text{IFWA3}} = \frac{\sum_{k=1}^{L} y_k^l}{2L},\ y_{\text{IFWS3}} = \sum_{k=1}^{L} y_k^l \text{ for } \alpha \to \infty,$$

$$y_k^l = \{y \mid w_l^{\mu} - w_l^{\nu} = \max_k (w_k^{\mu} - w_k^{\nu})\}\,. \tag{9}$$

Hereinafter, we refer to IFWA1, IFWA2 and IFWA3 as COA, BADD and MOM, respectively.

4 Methods for Adaptation in Intuitionistic Neuro-Fuzzy Network

In the process of IFIS initialization, cluster centers are found to construct the number N of if-then rules and the antecedents of the if-then rules. This can be carried out automatically using a subtractive clustering algorithm (SCA) (Chiu 1994). In that case, the number of clusters c is equal to the number of membership (non-membership) functions and, at the same time, to the number N of if-then rules. In the SCA, the potential P_k of data instance $\mathbf{x}_k$ is defined as follows

$$P_k = \sum_{j=1}^{n} e^{-\alpha \|\mathbf{x}_k - \mathbf{x}_j\|}\,, \tag{10}$$

where α=4/r_a^2 and r_a denotes the radius of influence. The radius of influence of a cluster is considered the most important parameter in establishing the number of cluster centers (and if-then rules). A large r_a results in fewer clusters, whereas a small r_a generates a large number of clusters and, thus, can lead to model over-fitting. The instance with the highest potential represents the cluster center of the first cluster. Then, an amount of potential from each data instance is subtracted as a function of its distance from the first cluster center

$$P_k \Leftarrow P_k - P_1^* e^{-\beta \|\mathbf{x}_k - \mathbf{x}_1^*\|^2}\,, \tag{11}$$

where $\mathbf{x}_1^*$ is the center of the first cluster, P_1^* is the potential of $\mathbf{x}_1^*$, and β=4/r_b^2. The positive constant r_b represents the radius defining the neighbourhood that will have measurable reductions in potential P_k.

In accordance with (Shing and Jang 1993), the adaptation of INFN (premise and consequent parameters) can be carried out, for example, using the following algorithms: (1) recursive LSE, where INFN is linearized w.r.t. the premise parameters, and extended Kalman filter is used to adapt all parameters (Jang 1991; Ramaswamy et al 1993; Wang 1998, Simon 2002); (2) Kaczmarz algorithm (Kaczmarz 1993; Strohmer and Vershynin

2007); (3) gradient descent algorithm with one pass of least-squares estimate (LSE), where LSE is applied to obtain the initial values of the consequent parameters, (Chiu 1994); and (4) PSO (Kennedy and Eberhart 1995; Shi and Eberhart 1998).

Let each input vector $\mathbf{x}_k^*$ be decomposed into two component vectors $\mathbf{y}_k^*$ and y_k^*, where $\mathbf{y}_k^*$ contains first n elements of $\mathbf{x}_k^*$ (input data) and y_k^* contains the output component. The output y is represented by a weighted average (weighted sum) of the output of each rule. Each cluster center $\mathbf{x}_k^*$ is considered an if-then rule. For the IFIS of Takagi-Sugeno type of the first order, i.e. $f(x_1,x_2, \ldots ,x_m)$ is a linear function, the y_k^* can be calculated as follows

$$y_k^*=\mathbf{G}_k\mathbf{y}_k+h_k, \tag{12}$$

where $\mathbf{G}_k$ is a constant vector and h_k is a constant. The estimation of the parameters of the given model can be understood as LSE in the form $\mathbf{AX}=\mathbf{B}$, where $\mathbf{B}$ is a matrix of output values, $\mathbf{A}$ is a constant matrix, and $\mathbf{X}$ is a matrix of parameters to be estimated.

Let P be the set of linear parameters and $\mathbf{X}$ be an unknown vector whose elements are parameters in P. Then, we seek the optimal solution of $\mathbf{X}$ using a LSE $\mathbf{X}^*$. In this process, the squared error $\|\mathbf{AX}-\mathbf{B}\|^2$ is minimized. Recursive LSE can be used to compute $\mathbf{X}^*$ effectively in the case of a low number of linear parameters (Jang 1993). Let $\mathbf{a}_i^T$ be the i-th input vector of matrix $\mathbf{A}$ and $\mathbf{b}_i^T$ be the i-th element of $\mathbf{B}$. Then $\mathbf{X}$ can be calculated as follows

$$\mathbf{X}_{i+1}=\mathbf{X}_i+\mathbf{S}_{i+1}\mathbf{a}_{i+1}(\mathbf{b}^T_{i+1}-\mathbf{a}^T_{i+1}\mathbf{X}_i),\ \mathbf{S}_{i+1}=1/\lambda[\mathbf{S}_i-(\mathbf{S}_i\mathbf{a}_{i+1}\mathbf{a}^T_{i+1}\mathbf{S}_i)/(\lambda+\mathbf{a}^{\mathrm{T}}_{i+1}\mathbf{S}_i\mathbf{a}_{i+1})], \tag{13}$$

where $\mathbf{S}_i$ is the covariance matrix, λ is the forgetting factor (effect of old data decay), and $\mathbf{X}^*=\mathbf{X}_n$. The initial conditions are $\mathbf{X}_0=0$ and $\mathbf{S}_0=\gamma\mathbf{I}$, where γ is a positive large number and $\mathbf{I}$ is the identity matrix. The LSE of $\mathbf{X}$ can be interpreted as the Kalman filter for the process (Jang 1991)

$$\mathbf{X}(k+1)=\mathbf{X}(k),\ \mathbf{Y}(k)=\mathbf{A}(k)\mathbf{X}(k)+e, \tag{14}$$

where e is noise, $\mathbf{X}(k)=\mathbf{X}_k$, $\mathbf{Y}(k)=\mathbf{b}_k$ and $\mathbf{A}(k)=\mathbf{a}_k$. Therefore, the recursive LSE presented above is referred to as the Kalman filter algorithm. Another method used to compute the LSE $\mathbf{X}^*$ is the Kaczmarz algorithm (Kaczmarz 1993) defined as follows

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \frac{\mathbf{b}_k - \mathbf{a}_k\mathbf{X}_k^T}{\mathbf{a}_k^T\mathbf{a}_k}\mathbf{a}_k\,. \tag{15}$$

Another possibility to adapt an IFIS is represented by gradient algorithms. Let the i-th error be defined as $E_i=(y_i-o_i)^2$, where y_i is the actual output and o_i is the predicted output. Then, the total error E is given as $E=\Sigma E_i$. The derivative of the overall error measure E with respect to a generic parameter α is

$$\frac{\partial E}{\partial \alpha} = \sum_{i=1}^{n}\frac{\partial E_i}{\alpha}\,. \tag{16}$$

The update formula for the generic parameter α is defined as

$$\Delta\alpha = -\eta \frac{\partial E}{\partial \alpha}, \text{ where } \eta = \frac{h}{\sqrt{\sum_{\alpha}\left(\frac{\partial E}{\partial \alpha}\right)^2}}, \tag{17}$$

where η is the learning rate, h is the step size and the length of each gradient transition in the parameter space. The value of h can be changed to vary the speed of convergence. Some practical difficulties associated with gradient descent are slow convergence and ineffectiveness at finding a good solution (Simon 2002). To overcome these limitations, PSO can be used to optimize both the antecedent and consequent parameters of the INFN.

PSO is a population based stochastic optimization method introduced by Kennedy and Eberhart (1995). Similar to other evolutionary computation methods like genetic algorithm (GA), the PSO is initialized with a population of random solutions (particles) in the search space. The PSO algorithm finds the global best solution (fitness) by adjusting the trajectory (velocity and position) of individual particle towards its best location and towards the best particle of the entire population. In other words, all the population is taken as the topological neighbors of the individual particle, which thus reflects information obtained from all the particles in the swarm (Shi and Eberhart 1998).

For a particle moving in a multidimensional search space, let $x_i(t)$ denotes the position of a particle i in search space, $v_i(t)$ denotes the velocity, and j is the dimension (the number of parameters to be optimized). Then the velocity and position of particle i can be calculated as

$$v_{i,j}(t+1) = \omega_i v_{i,j}(t) + c_1 r_{1,j}[p_{i,j}(t) - x_{i,j}(t)] + c_2 r_{2,j}[p_{g,j}(t) - x_{i,j}(t)], \tag{18}$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1), \tag{19}$$

where ω_i is the inertia weight for the i-th particle, c_1 and c_2 are constants (acceleration coefficients) used to scale the contribution of the cognitive and social components respectively, $r_{1,j}$ and $r_{2,j}$ are uniformly distributed random numbers in the range [0,1], $p_{i,j}(t)$ is the best position of the i-th particle remembered, and $p_{g,j}(t)$ is the best swarm position. Inertia weight ω_i provides a balance between global and local exploration and exploitation (Shi and Eberhart 1998). Linearly varying inertia weight ω_i over the generations has performed well in previous studies owing to faster convergence. We therefore used a linearly decreasing varying inertia weight $\omega_i(t+1)=\delta\omega_i(t)$, where δ is the inertia weight damping ratio. The particle velocity at any instant is usually limited to v_{max}.

In this study, we adopted the Pittsburgh approach known in the context of GA-based FISs, where each particle represents a full solution to the problem. In other words, each particle encodes a set of N if-then rules. The PSO was used to tune the parameters of both membership (non-membership) functions in the antecedent and the parameters of the linear functions in the consequent of the if-then rules. As a result, the dimension of a particle depends on both the problem's domain (the length of if-then rules) and on the number N of if-then rules, this is dimension d=N(4n+(n+1)). Fig. II depicts the encoding of if-then rules into a particle.

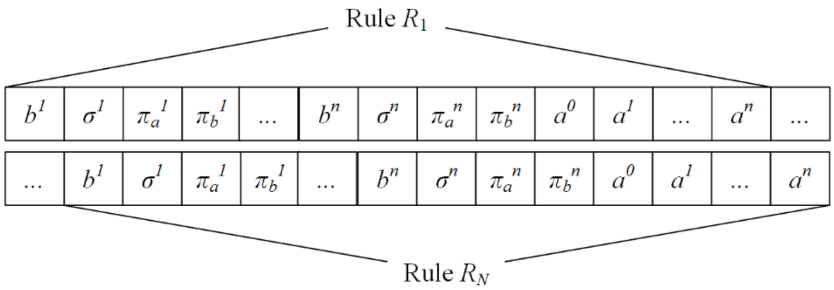

Fig. II Coding if-then rules into a particle

The position of each particle in the multidimensional search space changes depending on the particle's fitness values and the fitness value of its neighbours. Here, the fitness function used to evaluate a particle was RMSE.

Recently, PSO has been successfully applied to adapt FISs and their generalizations (Castillo and Melin 2012; Castillo et al. 2012; Chakravarty and Dash 2012) mainly due to its ability to quickly converge to an optimal solution and simplicity in adjusting PSO parameters.

5 Datasets

5.1 Credit Scoring Dataset

Credit scoring reduces information asymmetry between borrowers (issuers) and lenders (investors) by providing assessment of the credit risk of the borrowers. As output credit scores, we used the ratings provided by Standard and Poor's rating agency in 2011. We collected the ratings for 613 U.S. companies. The ratings AAA, AA+, AA, … ,C were transformed to credit scores 1,2,3, … ,21 (see Fig. III for their frequencies).

Recently, it has been demonstrated that credit scoring prediction can be performed based on textual analysis of documents related to assessed companies (Hájek and Olej 2013). Therefore, we first preprocessed textual data to obtain inputs to INFN and then we trained INFN to accurately predict the corresponding credit score. As a source of input data, we used the annual reports of the companies, which are freely available on the U.S. Securities and Exchange Commission EDGAR System (www.sec.gov/edgar.shtml). Specifically, we used only the Management Discussion and Analysis section of the annual reports, where management discusses past and present corporate performance. The textual part of the annual reports is considered as an important indicator of future financial performance (Hájek and Olej 2013).

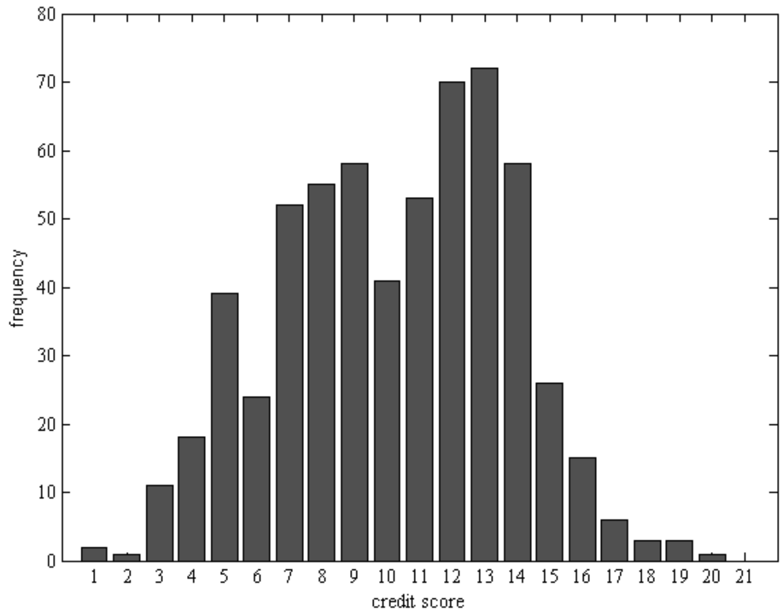

Fig. III Frequencies of corporate credit scores in the dataset

First, we preprocessed the documents using tokenization and lemmatization. Thus, potential term candidates were obtained. Second, the term candidates were compared with the positive (354 terms) and negative terms (2349 terms) developed specifically for financial domain (Loughran and McDonald 2011). Third, the *tf.idf* (term frequency–inverse document frequency) term weighting scheme was applied to calculate the importance of the terms obtained in the previous step. Fourth, the data were randomly divided into a training and testing set (4:1). This process was repeated five times. Fifth, a feature selection was performed using a correlation-based filter (Hall 1999). Note that feature selection was conducted after data division in order to prevent feature subset selection bias. On average, 29 terms were selected in training data sets. Table I shows all terms that were selected at least once.

Table I. A set of input attributes

Category	Terms
Positive	abl, adequ, advantag, benefici, boom, compliment, construct, favor, good, charit, ideal, inventor, leadership, loyal, plenti, prestigi, progress, prosper, rebound, satisfi, solv, stabl, strength, success, win
Negative	abolish, acquiesc, alien, anticompetit, bribe, catastroph, confront, deeper, deleteri, demis, deter, disagr, disclos, dishonest, disincent, divert, drastic,

	embargo, erron, exculp, exoner, expos, feloni, grossli, harsh, impair, imped, impound, inabl, incapacit, incompet, incomplet, inconclus, indict, insurrect, know, limit, malfunct, miscalcul, misdemeanor, nonpay, non-renew, object, obstacl, overdu, overestim, press, prevent, question, reck-less, refin, riskier, stagnant, stoppag, uncorrect, unfeas, unlicens, unreli, unremedi, unresolv, unsold, unwant, vulner, weak, worst, wrongdo

5.2 Benchmark Datasets

The following regression datasets were selected for the prediction using evolutionary INFN: Friedman benchmark function, Abalone, Auto MPG dataset, EDAT_1_1661, Forest fires, Machine CPU and Mortgage, for details see Table II. All datasets were obtained from the Keel datasets repository (Alcala-Fdez et al. 2011).

Table II. Description of benchmark regression datasets

	Friedman	Abalone	AutoMPG	EDAT_1_1661
Origin	Artificial	Real-world	Real-world	Real-world
Input variables	5	8	5	4
Real/integer	5/0	7/1	2/3	4/0
Instances	1200	4174	392	1655
	Forest fires	machineCPU	Mortgage	
Origin	Real-world	Real-world	Real-world	
Input variables	12	6	15	
Real/integer	7/5	0/6	15/0	
Instances	517	209	1049	

Friedman dataset is a synthetic benchmark dataset where the instances are generated using the following method. Generate the values of n=5 input variables, $x_1,x_2, \ldots ,x_5$ independently each of which uniformly distributed over [0.0, 1.0]. Obtain the value of the target variable y using the equation $y=10(\sin(\pi)x_1x_2)+20(x_3-0.5)^2+10x_4+5x_5+e$, where e is a Gaussian random noise $N(0,1)$. In the Abalone dataset, the age of abalone is predicted from physical measurements. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through. Other measurements are used to predict the age. The AutoMPG dataset concerns city-cycle fuel consumption in miles per gallon (Mpg), to be predicted in terms of 1 multi-valued discrete and 5 continuous attributes (two multi-valued discrete attributes (Cylinders and Origin) from the original dataset are removed). EDAT_1_1661 is a set of measurements of the light curve (time variation of the intensity) of the variable white dwarf star PG1159-035 during March 1989. A polynomial was used to normalize the signal to remove changes due to varying extinction (light absorption) and differing telescope properties. In the Forest fires dataset, the aim is to predict the burned area of forest fires, in the northeast region of Portugal, by using meteorological and other data. The aim of the machineCPU problem is to approximate the published relative performance of the

CPU using the following attributes: machine cycle time, minimum main memory, maximum main memory, cache memory, minimum channels in units and maximum channels in units. The Mortgage dataset contains the U.S. economic data from 01/04/1980 to 02/04/2000 on a weekly basis. The goal is to predict the 30 Year-Conventional Mortgage Rate using attributes such as maturity rates, money stock, savings deposits, etc.

6 Experimental Results

6.1 Credit Scoring using Text Information

As mentioned above, the identification of INFNs was carried out in two steps. In the first step, cluster centers were found using the SCA. Note that by using this algorithm, the number of cluster centers c is equal to the number of if-then rules N, $c=N$. Therefore, the radius of influence of a cluster r_a largely determines both the number of membership (non-membership) functions and the number of if-then rules. To control the complexity of the INFN (and the potential over-fitting risk), we examined the radiuses varying from the set $r_a=\{0.50,0.55, \dots ,0.95\}$. As a result, we obtained on average $c=N=5.00\pm0.63$ if-then rules (and membership and non-membership functions) for the five training sets. In the second step, the premise and consequent parameters of the INFN were optimized using (1) extended Kalman filter, (2) gradient descent with one pass of LSE, (3) Kaczmarz algorithm, and (4) PSO. The learning parameters of these algorithms were set as follows. The parameters of the Kalman filter were data forgetting factor (set to $\lambda=0.99$) and increasing factor of data forgetting factor (set to 0.99). The gradient algorithm was trained with the maximum number of epochs equal to 100, step size was set to $h=0.01$, step increasing rate to 1.1, and step decreasing rate to 0.9. The Kaczmarz algorithm was trained with the maximum number of sweeps set to 10. For the PSO, the setting of the parameters is presented in Table III.

The quality of prediction was measured by RMSE on testing data owing to the ordinal character of the credit scores. We conducted the experiments in Matlab Fuzzy Logic Toolbox.

Table III. Characteristics of the PSO parameters used in experiments

parameter	value
population	25
number of iterations	500
inertia weight ω_i	1
inertia weight damping ratio δ	0.99

personal learning coefficient c_1	1
global learning coefficient c_2	2
maximum particle velocity v_{max}	5

In the first set of experiments, we compared the defuzzification methods used in the INFNs, BADD (α=2) and MOM. Fig. IV shows that the MOM performed better than BADD in case of the Kalman learning algorithm. In contrast, BADD was an effective defuzzification method for the Kaczmarz learning algorithm. Overall, these findings suggest that the most accurate weighted consequent values were calculated using the functions with the maximum difference between w^{μ}_k and w^{ν}_k firing weights using PSO algorithm, in particular.

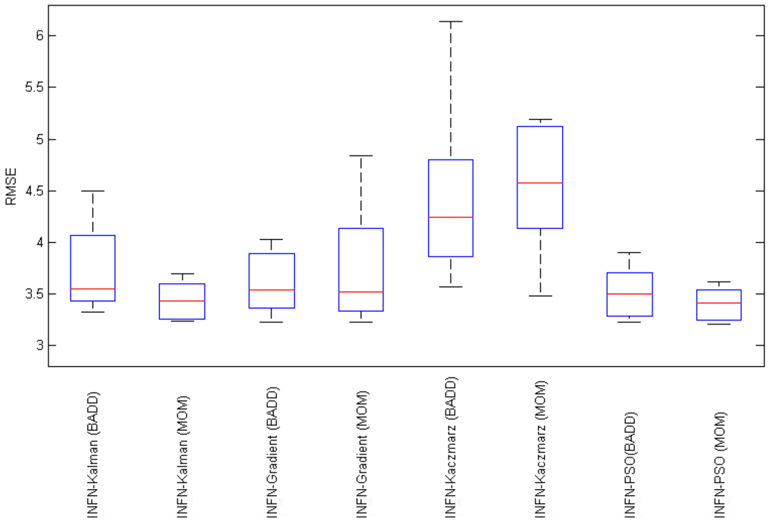

Fig. IV RMSE on testing data – comparison between INFN defuzzification methods. Median, lower, and upper quartile, and minimum and maximum RMSE are depicted in boxplots.

In the second set of experiments, we used the best results from the first set of experiments and compared them with traditional ANFIS algorithms, hybrid (combining LSE and backpropagation) and backpropagation. Hybrid and backpropagation algorithms are two commonly used algorithms to train ANFIS. The number of epochs was set to 10 for the hybrid algorithm and to 100 for the backpropagation algorithm. In addition, we trained the ANFIS using two evolutionary algorithms, GA and PSO. The setting of the PSO was the same as in the first set of experiments. The learning parameters of the GA were set as follows: population=25, epochs=500, crossover percentage=0.4, mutation percentage=0.7, mutation rate=0.15, and roulette wheel procedure was used for selection.

For the ANFIS, the initial setting of FIS was also determined using the SCA with the same procedure as for the INFN. Fig. V shows that the INFN trained with the Kalman, gradient and PSO algorithms performed better than ANFIS irrespective of the algorithm used.

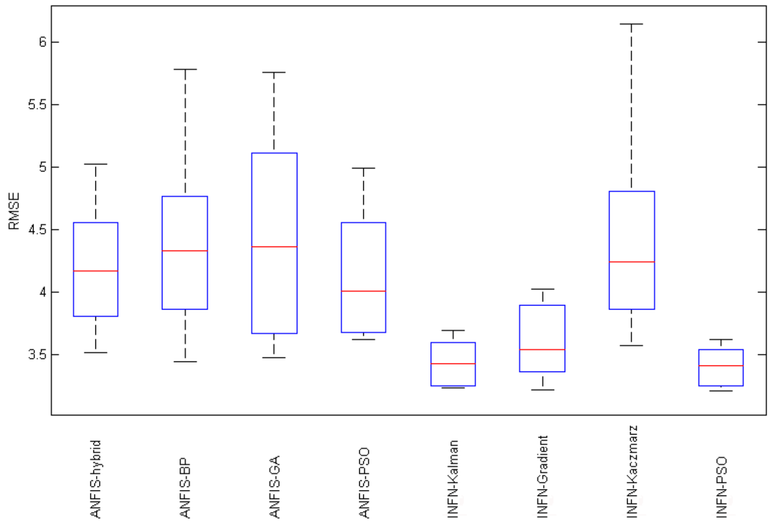

Fig. V RMSE on testing data – comparison between ANFIS and INFN. Median, lower, and upper quartile, and minimum and maximum RMSE are depicted in boxplots.

To test the significance of the results, we employed Student's paired *t*-tests (Table IV and Table V). The results show that INFN-PSO using the MOM as the defuzzification method performed statistically significantly better than both the ANFIS and INFN trained with other methods (Table V). On the other hand, the INFN-PSO (BADD) and INFN-Gradient (MOM) performed statistically similar as the INFN-PSO (MOM).

Table IV. Descriptive statistics of results for defuzzification methods and *p*-values for Student's paired *t*-tests vs. INFN-PSO (MOM)

	Mean±St.Dev.	*p*-value
INFN-Kalman (BADD)	3.753±0.424	0.071
INFN-Kalman (MOM)	3.435±0.176	0.068
INFN-Gradient (BADD)	3.609±0.291	0.045
INFN-Gradient (MOM)	3.770±0.580	0.172
INFN-Kaczmarz (BADD)	4.454±0.887	0.053
INFN-Kaczmarz (MOM)	4.538±0.616	0.010

INFN-PSO (BADD)	3.513±0.245	0.208
INFN-PSO (MOM)	3.402±0.154	-

Table V. Descriptive statistics of results for ANFIS vs. INFN and *p*-values for Student's paired *t*-tests vs. INFN-PSO

	Mean±St.Dev.	*p*-value
ANFIS-hybrid	4.203±0.504	0.013
ANFIS-BP	4.397±0.773	0.036
ANFIS-GA	4.444±0.820	0.038
ANFIS-PSO	4.144±0.506	0.028
INFN-Kalman	3.435±0.176	0.068
INFN-Gradient	3.609±0.291	0.045
INFN-Kaczmarz	4.454±0.887	0.053
INFN-PSO	3.402±0.154	-

6.2 Benchmark Datasets

The benchmark regression datasets were randomly divided into training and testing data in relation 4:1 and again this division was realized five times. To evaluate the prediction performance of the INFN-PSO, we compared it with the following algorithms: (1) ANFIS-hybrid, (2) ANFIS-BP (backpropagation), (3) ANFIS-GA, and (4) ANFIS-PSO. Hereinafter we report only the best results obtained either using BADD or MOM defuzzification method for the INFN-PSO. The quality of regression was measured by RMSE on testing data.

Again, the initial setting of the FISs and IFISs was conducted using SCA. Two sets of experiments were performed for the different numbers of if-then rules (and the numbers of membership and non-membership functions at the same time), N=3 and N=5, to demonstrate the effect of the complexity of FIS and IFIS learning parameters. We used Gaussian membership (and non-membership) functions for the FISs and IFISs, respectively. For the FISs, the minimum t-norm was selected.

In our experiments, we used the following settings of adaptation methods. The ANFIS-hybrid and ANFIS-BP algorithms were trained with the number of epochs set to 10 (and 500 for the BP), with step size h=0.01, step increasing rate set to 1.1, and step decreasing rate set to 0.9. The parameters of the ANFIS-GA were population (set to 25), epochs (500), crossover percentage (0.4), mutation percentage (0.7), and mutation rate (0.15). Again, roulette wheel selection was applied to choose individuals for crossover and mutation. In the training of the ANFIS-PSO, the parameters of the PSO was set as presented in Table III.

Tables VI and VII report the mean RMSE for the seven benchmark datasets, highlighting the best performing algorithm in italics. In general, the performances of the evaluated algorithms improved with an increasing number of if-then rules for two datasets, Friedman and ForestFires, suggesting higher complexity of these two tasks. In both cases, the INFN-PSO performed best for *N*=5.

In case of *N*=3 rules, the INFN-PSO performed best for the Friedman, Abalone, Auto-MPG and MachineCPU. Although the INFN-PSO showed generally promising results on all the datasets, the ANFIS-hybrid method performed better on the Mortgage dataset, suggesting that the PSO adaptation method may be ineffective for higher dimensional search spaces. An increase in the size of population may address this issue (Chen et al., 2015).

Table VI. Performance (mean RMSE ± St.Dev.) on benchmark datasets for *N*=3 rules

Dataset	ANFIS-hybrid	ANFIS-BP	ANFIS-GA	ANFIS-PSO	INFN-PSO
Friedman	1.865±0.329	1.722±0.251	2.305±0.188	1.650±0.120	*1.499±0.140*
Abalone	2.146±0.070	2.155±0.107	2.334±0.184	2.189±0.098	*2.139±0.080*
Auto-MPG	2.842±0.234	13.952±3.238	3.391±0.352	2.896±0.290	*2.817±0.287*
EDAT	*0.056±0.003*	4.375±2.239	*0.056±0.003*	0.057±0.003	0.057±0.003
ForestFires	65.29±28.00	63.57±28.25	*57.09±30.93*	57.75±32.99	58.35±32.99
MachineCPU	81.45±33.82	139.43±33.42	327.46±472.60	75.98±16.96	*66.17±21.17*
Mortgage	*0.083±0.006*	19.712±2.717	0.212±0.068	0.149±0.023	0.091±0.004

Table VII. Performance (mean RMSE ± St.Dev.) on benchmark datasets for *N*=5 rules

Dataset	ANFIS-hybrid	ANFIS-BP	ANFIS-GA	ANFIS-PSO	INFN-PSO
Friedman	1.602±0.095	1.450±0.050	2.090±0.109	1.506±0.130	*1.395±0.054*
Abalone	2.193±0.170	2.220±0.171	2.381±0.367	2.160±0.109	*2.112±0.071*
Auto-MPG	2.865±0.227	11.906±1.206	3.125±0.337	*2.750±0.304*	2.932±0.214
EDAT	0.062±0.011	2.680±2.137	0.059±0.006	0.059±0.007	*0.058±0.004*
ForestFires	553.77±951.91	675.13±761.31	59.17±31.54	55.22±32.14	*55.18±31.92*
MachineCPU	429.17±723.33	647.78±1088.47	88.38±39.63	91.59±46.90	*72.60±22.44*
Mortgage	*0.080±0.005*	18.402±3.100	0.329±0.174	0.120±0.035	0.087±0.004

Fig. VI illustrates the convergence of the INFN-PSO for two problems with a different complexity, Friedman dataset with *n*=5 and Mortgage dataset with *n*=15 input variables. To demonstrate the effect of particle complexity, we show the results for both *N*=3 and *N*=5 rules. We report average fitness values over the five training datasets. The INFN-PSO converges faster for the less complex Friedman dataset. However, the effect of *N* is limited, mainly due to the low number of rules (*N*=3 and *N*=5 rules, respectively). As indicated in Tables VI and VII, the population converges on a poor local optimum in case of the Mortgage dataset. Note also that the computation complexity of PSO increases exponentially as the dimensionality of the search space increases.

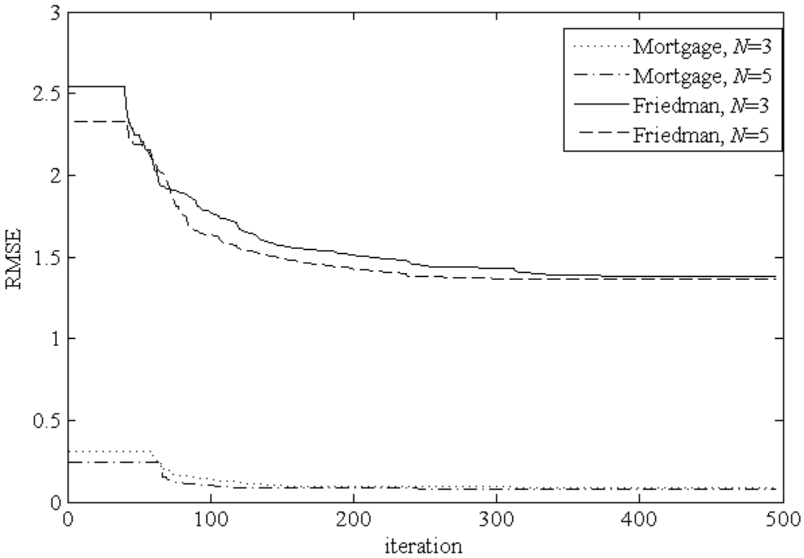

Fig. VI Convergence of INFN-PSO for Mortgage and Friedman datasets, *N*=3 and *N*=5 rules

To detect statistical differences in the prediction performance of the used methods, we performed nonparametric Friedman test because we could not guarantee the reliability of parametric tests. The Friedman test enables ranking of the algorithms according to the Friedman statistic. This test is a nonparametric analogue of the parametric two-way analysis of variance. The original results (RMSE) were converted to ranks. Average ranks were calculated in case of ties. The null hypothesis was tested which states that all the algorithms perform similarly (their ranks should be equal). The Friedman *p*-values obtained in Tables VIII and IX indicate the existence of significant differences between the evaluated algorithms. The lowest average ranking was achieved by the INFN-PSO for both tests, *N*=3 and *N*=5.

In the second step, we therefore used the INFN-PSO as a control algorithm in post-hoc procedures to determine which algorithms perform significantly worse. Specifically, we used the Holm and Finner post-hoc procedures which adjust the level of significance in a step-down manner (for details, see García et al. 2010). The results (*p*-values) are reported in Tables VIII a IX. Significant differences at $p<0.05$ are highlighted in bold.

The results of the post-hoc tests indicated that the INFN-PSO significantly outperformed most of the evaluated algorithms. In the case of less complex tasks (for *N*=3 rules and membership functions), only the ANFIS-hybrid did not perform significantly worse (Table VIII), whereas for *N*=5 all evaluated algorithms were outperformed by the INFN-PSO (Table IX).

Table VIII. Results of nonparametric tests (Friedman, Holm and Finner) for *N*=3 rules

N=3	Aver. ranking	Holm *p*-value	Finner *p*-value
ANFIS-hybrid	2.3429	0.0500	0.0500
ANFIS-BP	3.9429	**0.0125**	**0.0127**
ANFIS-GA	3.5429	**0.0167**	**0.0253**
ANFIS-PSO	2.9714	**0.0250**	**0.0377**
INFN-PSO	2.2000	-	-
Friedman *p*-value	$<10^{-5}$		

Table IX. Results of nonparametric tests (Friedman, Holm and Finner) for *N*=5 rules

N=5	Aver. ranking	Holm *p*-value	Finner *p*-value
ANFIS-hybrid	2.9143	**0.0250**	**0.0377**
ANFIS-BP	4.1714	**0.0125**	**0.0127**
ANFIS-GA	3.7286	**0.0167**	**0.0253**
ANFIS-PSO	2.2000	**0.0500**	**0.0500**
INFN-PSO	1.9857	-	-
Friedman *p*-value	$<10^{-5}$		

7 Conclusion

Taken together, we have demonstrated that INFNs trained by PSO algorithm may significantly outperform ANFIS. In the case study of credit scoring using text information, hybrid and backpropagation algorithms were employed as two common algorithms to train ANFIS. The MOM defuzzification method, proposed in this study, has also shown promising results for use in the output layer of the INFN.

This class of INFNs adapted by PSO appears to provide better performance especially in the cases where high uncertainty and imprecision have to be captured in membership functions' design. Financial prediction seems to be a suitable application domain. Previous evidence with type-2 FISs supports this finding (Huarng and Yu 2005; Zarandi et al. 2009; Bernardo et al. 2013).

Consistent with previous studies, we show that the introduction of hesitation part may improve the prediction performance of FISs. However, substantially more experiments should be conducted to generalize our findings, particularly on higher dimensional prediction problems. Future research should also concentrate on the optimization of the base of if-then rules. Alternative implementations of the PSO algorithm should also be considered like, for example, the PSO with a leader and followers (Wang and Wang, 2008). Finally, we recommend further extensions of INFNs corresponding to recent development in IF-sets, for instance temporal IF-sets (Chen and Tu 2015).

The experiments in this study were carried out in Fuzzy Logic Toolbox, Matlab 2010b using the MS Windows 7 operation system.

Acknowledgments. This work was supported by the scientific research project of the Czech Sciences Foundation Grant No: 13-10331S.

References

[1] Akram MS, Habib S, Javed I (2014) Intuitionistic fuzzy logic control for washing machines. Indian J Sci Technol 7(5):654-661

[2] Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, Herrera F (2011) KEEL Data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. J Multiple-Valued Logic and Soft Comput 17(2-3):255-287

[3] Angelov P (1995) Crispification: Defuzzification over intuitionistic fuzzy sets. BUSEFAL 64:51-55

[4] Angelov P (2001) Multi-objective optimisation in air-conditioning systems: Comfort/discomfort definition by IF sets. Notes on Intuitionistic Fuzzy Sets 7(1):10-23

[5] Angelov P (2012) Evolving fuzzy systems. In: Computational complexity: Theory, techniques, and applications, Springer-Verlag, Berlin, pp 1053-1065

[6] Atanassov KT (1986) Intuitionistic fuzzy sets. Fuzzy Set and Syst 20:87-96

[7] Atanassov KT (1999) Intuitionistic fuzzy sets. Physica-Verlag, Heidelberg New York

[8] Barrenechea E (2009) Generalized Atanassov’s intuitionistic fuzzy index. Construction method. In: IFSA-EUSFLAT, Lisbon, pp 478-482

[9] Bernardo D, Hagras H, Tsang E (2013) A genetic type-2 fuzzy logic based system for the generation of summarised linguistic predictive models for financial applications. Soft Comput 17(12):2185-2201

[10] Castillo O, Alanis A, Garcia M, Arias H (2007) An intuitionistic fuzzy system for time series analysis in plant monitoring and diagnosis. Appl Soft Comput 7(4):1227-1233

[11] Castillo O, Martínez-Marroquín R, Melin P, Valdez F, Soria J (2012) Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. Inform Sciences 192:19-38

[12] Castillo O, Melin P (2012) Optimization of type-2 fuzzy systems based on bio-inspired methods: A concise review. Inform Sciences 205:1-19

[13] Chakravarty S, Dash PK (2012) A PSO based integrated functional link net and interval type-2 fuzzy logic system for predicting stock market indices. Appl Soft Comput 12(2):931-941

[14] Chen S, Montgomer J, Bolufé-Röhler A (2015) Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution. Appl Intell 42(3):514-526

[15] Chen LH, Tu CC (2015) Time-validating-based Atanassov's intuitionistic fuzzy decision-making. IEEE T Fuzzy Syst 23(4):743-756

[16] Chiu S (1994) Fuzzy model identification based on cluster Eestimation. J Intell Fuzzy Syst 2:267-278

[17] Demertzis K, Iliadis L, Avramidis S, El-Kassaby YA (2016) Machine learning use in predicting interior spruce wood density utilizing progeny test information. Neural Comput Appl. doi: 10.1007/s00521-015-2075-9

[18] Deschrijver G, Cornelis C, Kerre E (2004) On the representation of intuitionistic fuzzy t-norm and t-conorm. IEEE T Fuzzy Syst 12:45-61

[19] Dubois D, Prade H (2005) Interval-valued fuzzy set, possibility theory and imprecise probability. In: European Society for Fuzzy Logic and Technology, EUSFLAT/LFA, Barcelona, Spain, pp 314-319

[20] García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Inform Sciences 180(10):2044-2064

[21] Hagras H, Wagner Ch (2012) Towards the widespread use of type-2 fuzzy logic systems in real world applications. IEEE Comput Intell Mag 7(3):4-24

[22] Hájek P (2012) Credit rating analysis using adaptive fuzzy rule-based systems: An industry specific approach. Cent Eur J Oper Res 20(3):421-434

[23] Hájek P, Olej V (2012) Adaptive Intuitionistic Fuzzy Inference Systems of Takagi-Sugeno Type for Regression Problems. In: Iliadis LS, Maglogianis I, Papadopoulos H (ed) Springer, Heidelberg Dordrecht London New York pp 206-216

[24] Hájek P, Olej V (2014) Defuzzification methods in intuitionistic fuzzy inference systems of Takagi-Sugeno type. The case of corporate bankruptcy prediction. In: Fuzzy Systems and Knowledge Discovery (FSKD'14), Xiamen, China, pp 240-244

[25] Hájek P, Olej V (2013) Evaluating sentiment in annual reports for financial distress prediction using neural networks and support vector machines. In: Iliadis, Papadopoulos H, Jayne Ch (ed) Springer-Verlag Berlin Heidelberg pp 1-10

[26] Hall MA (1999) Correlation-based feature selection for machine learning. Dissertation, The University of Waikato

[27] Henzgen S, Strickert M, Hüllermeier E (2014) Visualization of evolving fuzzy rule-based systems. Evolving Systems 5(3):175-191

[28] Huarng K, Yu HK (2005) A type-2 fuzzy time series model for stock index forecasting. Stat Mech Appl 353:445-462

[29] Jang JSR (1991) Fuzzy modeling using generalized neural networks Kalman filter algorithm. In: Artificial Intelligence (AAAI-91), pp 762-767

[30] Jang JSR (1993) ANFIS: Adaptive-network-based fuzzy inference systems. IEEE T Syst Man Cyb 23(3):665-685

[31] Kaczmarz S (1993) Approximate solution of systems of linear equations. Int J Control 53:1269-1271

[32] Kasabov N (2015) Evolving connectionist systems: From neuro-fuzzy-, to spiking- and neuro-genetic. In: Springer handbook of computational intelligence, Springer-Verlag, Berlin Heidelberg, pp 771-782

[33] Kennedy J, Eberhart R (1995) Particle swarm optimization. In: IEEE International Conference on Neural Networks, pp 1942-1948. doi: 10.1109/ICNN.1995.488968

[34] Klement EP, Mesiar R, Pap E (2004) Triangular norms. Position paper I: Basic analytical and algebraic properties. Fuzzy Set Syst 143:5-26
[35] Liang Q, Mendel JM (2000) Interval type-2 fuzzy logic systems: Theory and design. IEEE T Fuzzy Syst 8(5):535-550
[36] Loganathan C, Girija KV (2013) Hybrid learning for adaptive neuro fuzzy inference system. Int J Eng Sci 2(11):6-13
[37] Loughran T, McDonald B (2011) When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. J Financ 66(1):35-65
[38] Maciel L, Lemos A, Gomide F, Ballini R (2012) Evolving fuzzy systems for pricing fixed income options. Evolving Systems 3(1):5-18
[39] Mendel JM (2006) Interval type-2 fuzzy logic systems made simple. IEEE T Fuzzy Syst 14(6):808-821
[40] Olej V, Hájek P (2010) IF-inference systems design for prediction of ozone time series: The case of Pardubice micro-region. In: Diamantaras K, Duch W, Iliadis LS (ed) Springer, Berlin Heidelberg New York pp 1-11
[41] Olej V, Hájek P (2011) Comparison of fuzzy operators for IF-inference systems of Takagi-Sugeno type. In: Iliadis LS, Maglogianis I, Papadopoulos H (ed) Springer, Berlin Heidelberg New York pp 92-97
[42] Ramaswamy P, Riese M, Edwards RM, Lee KY (1993) Two approaches for automating the tuning process of fuzzy logic controllers. In: IEEE Conf. on Decision and Control, San Antonio, TX, pp 1753-1758
[43] Shi Y, Eberhart RC (1998) A modified particle swarm optimizer. In: IEEE International Conference on Evolutionary Computation, pp 69-73
[44] Shing J, Jang R (1993) ANFIS: Adaptive network based fuzzy inference system. IEEE T Syst Man Cyb 23(3):665-685
[45] Simon D (2002) Training fuzzy systems with the extended Kalman filter. Fuzzy Set Syst 132(2):189-199
[46] Strohmer T, Vershynin R (2007) A randomized Kaczmarz algorithm with exponential convergence. J Fourier Anal Appl 15(2):262-278
[47] Wang J, Wang D (2008) Particle swarm optimization with a leader and followers. Progress in Nat Sci 18(11):1437-1443
[48] Wang L, Ye J (1998) Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter. Fuzzy Set Syst 101:353-362
[49] Zarandi F, Rezaee B, Turksen IB, Neshat E (2009) A type-2 fuzzy rules-based expert system model for stock price analysis. Expert Syst Appl 36:139-154