Tsinghua Science and Technology

Volume 20 | Issue 2 Article 5

2015

# Adaptive Electric Load Forecaster

Mingchui Dong
*the Department of Electrical and Computer Engineering, University of Macau, Macao 999078, China.*

Chinwang Lou
*the Department of Electrical and Computer Engineering, University of Macau, Macao 999078, China.*



## Recommended Citation

Mingchui Dong, Chinwang Lou. Adaptive Electric Load Forecaster. Tsinghua Science and Technology 2015, 20(2): 164-174.

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 05/11 pp164-174
Volume 20, Number 2, April 2015

# Adaptive Electric Load Forecaster

Mingchui Dong* and Chinwang Lou

**Abstract:** In this paper, a methodology, Self-Developing and Self-Adaptive Fuzzy Neural Networks using Type-2 Fuzzy Bayesian Ying-Yang Learning (SDSA-FNN-T2FBYYL) algorithm and multi-objective optimization is proposed. The features of this methodology are as follows: (1) A Bayesian Ying-Yang Learning (BYYL) algorithm is used to construct a compact but high-performance system automatically. (2) A novel multi-objective T2FBYYL is presented that integrates the T2 fuzzy theory with BYYL to automatically construct its best structure and better tackle various data uncertainty problems simultaneously. (3) The weighted sum multi-objective optimization technique with combinations of different weightings is implemented to achieve the best trade-off among multiple objectives in the T2FBYYL. The proposed methods are applied to electric load forecast using a real operational dataset collected from Macao electric utility. The test results reveal that the proposed method is superior to other existing relevant techniques.

**Key words:** load forecaster; Bayesian Ying-Yang learning algorithm; type-2 fuzzy theory

## 1 Introduction

In 21st century, due to the advocacy of green cities and zero carbon emission, many renewable energy-related applications such as local wind farms, photovoltaic cells, battery storage, electric vehicles, and various sensors are widely installed and distributed at the customer side. Therefore, electric load demand is foreseen to have larger uncertainty, indefiniteness, and variability due to the installation of diversified renewable energy applications, stimulation of energy efficiency, and promotion of demand response programs. However, existing techniques of load forecasting, which focus only on forecasting accuracy, would be no longer adequate. New forecasting techniques with advanced high-level abilities such as self-learning and self-adaptability are required for responding quickly to various changes in an environment and for tackling uncertainties in future smart power grids.

To this end, a novel electric load forecasting technique Self-Developing and Self-Adaptive Fuzzy Neural Networks using multi-objective Type-2 Fuzzy Bayesian Ying-Yang Learning (SDSA-FNN-T2FBYYL) algorithm based on the Type-2 (T2) fuzzy theory[1–6] and the Bayesian Ying-Yang Learning (BYYL) algorithm is proposed.

The features of this technique are highlighted below: (1) Unlike traditional methodologies in which a structure of FNN system is predefined[7–9] with prior expert knowledge, BYYL is adopted in this study for self-learning from data to find and construct a compact but high-performance system automatically. (2) To ensure that the electric load forecasting technique can tackle various data uncertainty problems with ease, a novel methodology integrating the T2 fuzzy theory with BYYL is proposed, in which crisp variables of the original BYYL are represented by the proposed fuzzy variables in three-dimensional space, thereby offering an additional degree of freedom to model data uncertainty, while other existing relevant

• Mingchui Dong and Chinwang Lou are with the Department of Electrical and Computer Engineering, University of Macau, Macao 999078, China. E-mail: mcdong@umac.mo; jerrylcw2010@macau.ctm.net.
* To whom correspondence should be addressed.
Manuscript received: 2012-12-17; revised: 2014-07-07; accepted: 2014-11-15

techniques do not. (3) The integration results in one multi-objective T2FBYYL. The weighted sum multi-objective optimization technique with combinations of different weightings is implemented achieving the best trade-off among multiple objectives in T2FBYYL.

The proposed methods are applied to electric load forecasting using a real operational dataset collected from Macao electric utility. The test results indicate that the proposed method performs better than other existing relevant techniques.

## 2 Review of T2 Fuzzy Set Theory

Some key definitions pertaining to T2 fuzzy sets[1], which are used in this study, are listed below.

**Definition 1**. T2 fuzzy set in point-value representation, Eq. (1).

As shown in Fig. 1, one T2 fuzzy set $\tilde{A}$ is characterized by the T2 Membership Function (MF) $\mu_{\tilde{A}}(v,u)$, in which $0 \leqslant \mu_{\tilde{A}}(v,u) \leqslant 1$,

$$\tilde{A} = \{((v,u), \mu_{\tilde{A}}(v,u)) | \forall v \in X, \forall u \in J_v \subseteq [0,\ 1]\} \tag{1}$$

where $v$ is the primary variable of one T2 fuzzy set, $v \in X$ ($X$ is the input space). $u$ is the secondary variable of one T2 fuzzy set, in which $u \in J_v$. $J_v$ is the primary membership of $v$, in which $J_v \subseteq [0,\ 1]$. $\mu_{\tilde{A}}(v,u)$ is the secondary grade at point $(v,u)$. $\tilde{A}$ can be expressed using Eq. (2) as well, where $\int\int$ denotes union over all admissible $v$ and $u$.

$$\tilde{A} = \int_{v\in X}\int_{u\in J_v} \mu_{\tilde{A}}(v,u)/(v,u),\ J_v \subseteq [0,1] \tag{2}$$

**Definition 2**. Interval T2 fuzzy set, Eq. (3).

When all $\mu_{\tilde{A}}(v,u) = 1$ in Eq. (2), then $\tilde{A}$ is an Interval T2 (IT2) fuzzy set.

$$\tilde{A} = \int_{v\in X}\int_{u\in J_v} 1/(v,u),\ J_v \subseteq [0,1] \tag{3}$$

**Definition 3**. Footprint Of Uncertainty (FOU), Eq. (4).

Uncertainty in the primary memberships of an IT2 fuzzy set consists of a bounded region called the footprint of uncertainty. It is the union of all primary memberships.

$$\mathrm{FOU}(\tilde{A}) = \bigcup_{v\in X} J_v \tag{4}$$

**Definition 4**. Upper Membership Function (UMF), Eq. (5) and Lower Membership Function (LMF), Eq. (6).

UMF is associated with the upper bound of FOU ($\tilde{A}$), and it is denoted as $\overline{\mu}_{\tilde{A}}(v), \forall v \in X$. LMF is associated with the lower bound of FOU($\tilde{A}$), and it is denoted as $\underline{\mu}_{\tilde{A}}(v), \forall v \in X$.

$$\overline{\mu}_{\tilde{A}}(v) \equiv \overline{\mathrm{FOU}(\tilde{A})}, \forall v \in X \tag{5}$$

$$\underline{\mu}_{\tilde{A}}(v) \equiv \underline{\mathrm{FOU}(\tilde{A})}, \forall v \in X \tag{6}$$

## 3 BYYL Algorithm

A BYY system[10–18] characterizes each observation $z \in X$ and its corresponding inner representation $y \in Y$ via the following Bayesian decomposition:

$$p(z,y) = p(z)p(y|z) \tag{7}$$

$$q(z,y) = q(y)q(z|y) \tag{8}$$

Equations (7) and (8) are called Yang and Ying machines, respectively. They are not equal unless $y$ is the optimal solution. However, under the principle of Ying-Yang harmonization, their difference should be minimized.

Given a dataset $D_z = \{z_i\}_{i=1}^N$, where $N$ is the number of input data, learning a task on a BYY system involves determining the probability densities $p(y|z)$, $p(z)$, $q(z|y)$, and $q(y)$ through a harmony learning principle implemented by maximizing the following harmony function,

$$H(p||q) = \int (y|z)\,p(z)\log[q(z|y)q(y)]\mathrm{d}z - \log z_q \tag{9}$$

where $z_q$ is a regularization term[12, 13].

For modeling one Gaussian mixture, we can use the following structure:

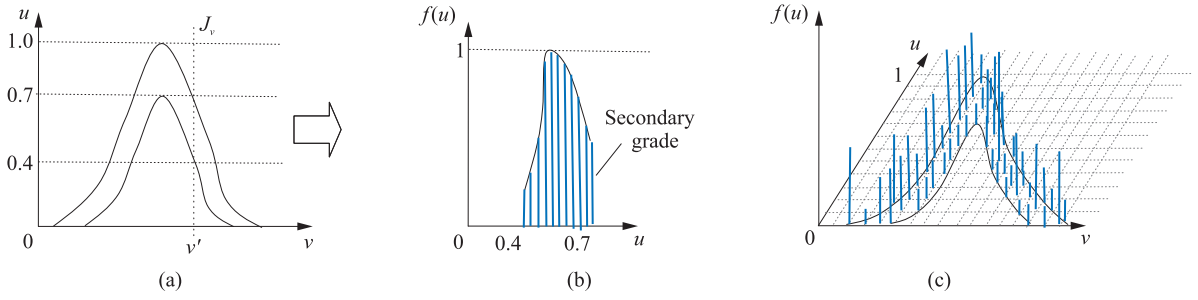


**Fig. 1 (a) One T2 fuzzy set projected onto $v-u$ plane. (b) Secondary MF (vertical slice) at $v'$. (c) T2 fuzzy set represented in three-dimensional space (bars are secondary grades in three-dimensional space).**

$$q(y)=\sum_{j=1}^{k}\alpha_j\delta(y-j),\alpha_j\geqslant 0,$$

$$\sum_{j=1}^{k}\alpha_j=1,y=1,\cdots,k \tag{10}$$

$$p(z)=p_0(z) \tag{11}$$

$$q(z|y=j)=G(z|\boldsymbol{\theta}_j)=G(z|\mu_j,\Sigma_j)=$$

$$\frac{1}{(2\pi)^{1/n}|\Sigma_j|^{1/2}}\exp\left(-\frac{1}{2}(z-\mu_j)\Sigma_j^{-1}(z-\mu_j)\right) \tag{12}$$

$$p(y|z)\geqslant 0,\quad \sum_{y=1}^{k}p(y|z)=1 \tag{13}$$

$$z_q=1 \tag{14}$$

where $\alpha_j$ is one discrete probability distribution; $k$ is the structural scale; $p_0(z)$ is the empirical density given by $p_0(z)=\dfrac{1}{N}\sum_{i=1}^{N}\delta(z-z_i)$; $\delta(\cdot)$ is Kronecker function; $G(z|\mu_j,\Sigma_j)$ is Gaussian density function with mean $\mu_j$ and variance $\Sigma_j$.

After substituting Eqs. (10)-(14) into Eq. (9) and considering that $N$ is sufficiently large, we have the following:

$$H(p||q)=J(\boldsymbol{\Theta}_k)=$$

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{y=1}^{k}p(y|\,z_i)\log\alpha_y G(z_i|\mu_y,\Sigma_y) \tag{15}$$

where $\boldsymbol{\Theta}_k=\{\alpha_y,\mu_y,\Sigma_y,p(y|z_i),\ i=1,\cdots,N\}_{y=1}^{k}$.

Global maximization of Eq. (15) will yield each probability density $p(y|z_i)$ as follows:

$$p(y|z_i)=\begin{cases}1, & \text{if } y=\arg\max\limits_j \alpha_j q(z_i|\mu_j,\Sigma_j);\\ 0, & \text{otherwise}\end{cases} \tag{16}$$

Equation (16) indicates that the correct number of Gaussians will be retained in this Gaussian mixture to match the actual number, while the extra Gaussians will be attenuated to zero. In other words, automatic detection of the correct number of Gaussians is achieved.

## 4 System Structure of SDSA-FNN-T2FBYYL

The proposed SDSA-FNN-T2FBYYL system is based on the Takagi-Sugeno Fuzzy Logic System (TS FLS), which can approximate any real-occurring nonlinear relationship to a certain degree of accuracy[19, 20]. The system structure shown in Fig. 2 is characterized by the following rule structure.

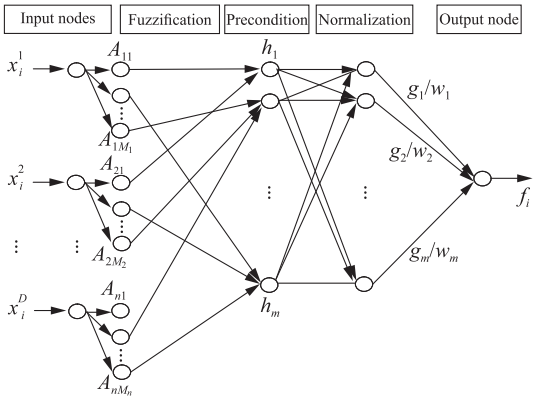


**Fig. 2 Structure of SDSA-FNN-T2FBYYL.**

Rule $l$: If $x_i^1$ is in $A_{l1}$ and $x_i^2$ is in $A_{l2}$ and $\cdots$ and $x_i^D$ is in $A_{lD}$, $f_l$ is $g_l$ with $w_l$.

Where $\boldsymbol{x}_i$ is the input data vector $[x_i^1,x_i^2,\cdots,x_i^D]^{\mathrm{T}}$; $f_l$ is the output of $l$-th fuzzy rule; $A_{lj}$ is the fuzzy set with $j$-th input variable $x_i^j$ in the $l$-th rule ($l$= 1, 2, $\cdots$, $m$, $j$= 1, 2, $\cdots$, $D, i=1,\cdots,N$); $g_l$ is $l$-th fuzzy rule's weight between the normalization node and the output node; $w_l$ is $l$-th fuzzy rule's confidence factor between the normalization node and the output node; $m$ and $D$ are the number of fuzzy rules and the number of input variables, respectively.

The complete input/output relation of the system can be formulated as follows (Remark: $I$ is the node input, $O$ is the node output, subscript and superscript represent node and layer index, respectively):

(1) Layer 1—Input nodes

As described below, each node transmits the input vector $\boldsymbol{x}_i=[x_i^1,\cdots,x_i^D]^{\mathrm{T}}$ to the next layer directly.

$$I_j^1=x_j,\ O_j^1=I_j^1 \tag{17}$$

where $j$= 1, 2, $\cdots$, $D$.

(2) Layer 2—Fuzzification nodes

In this layer, a novel fuzzification scheme based on T2FBYYL is used for obtaining the fuzzy linguistic labels on the input data vector. Each neuron represents a cluster. The output of this node specifies the degree to which an input $x_i^j$ belongs to $k$-th fuzzy set of the associated input node $j$ in layer 1.

$$I_{jk}^2=x_j,\ O_{jk}^2=A_{jk}=\exp\left[-\frac{(x_i^j-\mu_{jk})^2}{\sigma_{jk}^2}\right] \tag{18}$$

where $\mu_{jk}$ and $\sigma_{jk}$ are the center and width of this linguistic label, respectively, $j$= 1, 2, $\cdots$, $D$, $k$= 1, 2, $\cdots$, $M_j$, and $M_j$ is the number of fuzzy sets on the $j$-th input variable.

Details of the novel fuzzification scheme are given in Section 5.

(3) Layer 3—Precondition nodes

Each node in this layer represents one fuzzy rule and performs antecedent matching this rule.

$$I_l^3 = A_{l1}A_{l2}\cdots A_{ln},\ O_l^3 = h_l = I_l^3 \tag{19}$$

where $l$= 1, 2, $\cdots$, $m$ ($m$ is the number of fuzzy rules and $m = M_1 \times M_2 \times \cdots \times M_n$).

(4) Layer 4—Normalization nodes

The number of nodes in this layer is equal to that in layer 3. Each firing strength value calculated in layer 3 is normalized in this layer.

$$I_l^4 = h_l / \sum_{l=1}^{m} h_l,\ O_l^4 = I_l^4 \tag{20}$$

(5) Layer 5—Output node

Node in this layer generates the system output. The function of this layer is to combine the incoming signals from layer 4 and produce an output response $f_i$ ($i$-th input data vector generates $i$-th output).

$$f_i = \left(\sum_{l=1}^{m} g_l h_l w_l\right) / \left(\sum_{l=1}^{m} h_l w_l\right) \tag{21}$$

## 5 Fuzzification Using T2 Fuzzy BYYL

Fuzzification is a process related to the automatic determination of fuzzy linguistic labels for each dimension of an input data vector $\boldsymbol{x}_i$. In its early days, the fuzzification was implemented by grid-based partition[21]. Then clustering analysis by offline or online incremental process[22–27] dominated gradually. In this study, T2FBYYL is used for the fuzzification. The complete mathematical derivation is presented below.

If a Gaussian membership function is used, the joint probability density of the $j$-th dimension of the $i$-th input data $\boldsymbol{x}_i$ is calculated as follows:

$$p(x_i^j, \tilde{\boldsymbol{\Theta}}_{k^j}^j) = \sum_{y=1}^{k^j} \alpha_y^j G(x_i^j | \tilde{\boldsymbol{\theta}}_y^j) \tag{22}$$

where $k^j$ is the number of Gaussians in the $j$-th dimension of the $i$-th input data vector; $G(x_i^j|\tilde{\boldsymbol{\theta}}_y^j)$ is Gaussian density function with parameters $\tilde{\boldsymbol{\theta}}_y^j = \{\tilde{\mu}_y^j, \tilde{\sigma}_y^j\}$; $\tilde{\boldsymbol{\Theta}}_{k^j}^j$ is the set of model parameters and $\tilde{\boldsymbol{\Theta}}_{k^j}^j = \{\tilde{\alpha}_y^j, \tilde{\boldsymbol{\theta}}_y^j\}_{y=1}^k$; $\tilde{\mu}_y^j$ and $\tilde{\sigma}_y^j$ are T2 fuzzy mean and variance of $y$-th cluster in the $j$-th dimension.

Fuzzification using T2FBYYL involves two learning phases, i.e., parameter learning $\tilde{\boldsymbol{\Theta}}_{k^j}^j$ and determination of the correct number of clusters $k^j$ in each dimension of the $i$-th input data vector $\boldsymbol{x}_i$. As described in Section 3, this task can be implemented by maximizing Eq. (15) globally.

However, as pointed out in Refs. [10, 13, 14], the process of maximizing Eq. (15) can get trapped in a local maximum easily. To overcome this drawback, one simulated annealing BYYL is adopted in this study. The main difference of our approach from those in Refs. [16, 17] is that the T2 fuzzy theory is integrated with BYYL, as explained below.

In the light of Eq. (60) in Ref. [14], Eq. (15) can be reformulated by augmenting the parameter $\lambda$. This parameter is used to guide the searching direction for obtaining the globally optimum solution, i.e., after integrating parameter $\lambda$ and the T2 fuzzy theory, Eq. (15) becomes

$$L_\lambda(\tilde{p}(y^j|x^j), \tilde{\boldsymbol{\Theta}}_{k^j}^j) = J(\boldsymbol{\Theta}_{k^j}^j) + \lambda O_N(\tilde{p}(y^j|x^j)),\quad j = 1, \cdots, D \tag{23}$$

where

$$O_N(\tilde{p}(y^j|\boldsymbol{x}^j)) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{y=1}^{k^j} \tilde{p}(y^j|x_i^j)\log\tilde{p}(y^j|x_i^j) \tag{24}$$

The parameters $(\tilde{p}(y^j|\boldsymbol{x}^j), \tilde{\boldsymbol{\Theta}}_{k^j}^j)$ can be separated into two groups, i.e., $\boldsymbol{\Theta}_1 = \{\tilde{p}(y^j|x_i^j), i = 1, \cdots, N, y = 1, \cdots, k\}$ and $\tilde{\boldsymbol{\Theta}}_2 = \{\tilde{\alpha}_y^j, \tilde{\mu}_y^j, \tilde{\sigma}_y^j\}$ such that Eq. (5) becomes

$$L_\lambda(\tilde{p}(y^j|\boldsymbol{x}^j),\ \tilde{\boldsymbol{\Theta}}_{k^j}^j) = L_\lambda(\tilde{\boldsymbol{\Theta}}_1, \tilde{\boldsymbol{\Theta}}_2) \tag{25}$$

Equation (25) can be maximized using the following alternative iterative procedure.

**Step 1**: Fix $\tilde{\boldsymbol{\Theta}}_2 = \tilde{\boldsymbol{\Theta}}_2^{\text{old}}$, get $\tilde{\boldsymbol{\Theta}}_1^{\text{new}} = \arg\max_{\tilde{\Theta}_1} L_\lambda(\tilde{\boldsymbol{\Theta}}_1,\ \tilde{\boldsymbol{\Theta}}_2)$.

**Step 2**: Fix $\tilde{\boldsymbol{\Theta}}_1 = \tilde{\boldsymbol{\Theta}}_1^{\text{old}}$, get $\tilde{\boldsymbol{\Theta}}_2^{\text{new}} = \arg\max_{\tilde{\Theta}_2} L_\lambda(\tilde{\boldsymbol{\Theta}}_1,\ \tilde{\boldsymbol{\Theta}}_2)$.

## 6 Multi-objective T2 Fuzzy BYYL

The detailed mathematical derivation of the parameter learning ($\tilde{\boldsymbol{\Theta}}_1$ and $\tilde{\boldsymbol{\Theta}}_2$) described in Section 5 is presented below.

**Step 1**: Fix $\tilde{\boldsymbol{\Theta}}_2$ and solve $\tilde{\boldsymbol{\Theta}}_1$ by maximizing Eq. (25) with respect to $\tilde{\boldsymbol{\Theta}}_1$.

Parameter learning $\tilde{p}(y|x_i^j)$

$$\text{Max}\ \ L_\lambda(\tilde{p}(y|\boldsymbol{x}^j), \tilde{\boldsymbol{\Theta}}_{k^j}^j) = \frac{1}{N}\sum_{i=1}^{N}\sum_{y=1}^{k^j} \tilde{p}(y|x_i^j)\log\tilde{\alpha}_y\tilde{p}(x_i^j|\boldsymbol{\theta}_y^j) - \lambda\frac{1}{N}\sum_{i=1}^{N}\sum_{y=1}^{k^j}\tilde{p}(y|x_i^j)\log\tilde{p}(y|x_i^j) \tag{26}$$

s.t.

$$\sum_{y=1}^{k} \tilde{p}(y|x_i^j) = 1.$$

First, each $\tilde{p}(y|x_i^j)$ and $\tilde{\alpha}_y$ are represented by interval T2 fuzzy set as below,

$$\tilde{p}(y|x_i^j)=[\overline{p}(y|x_i^j), \underline{p}(y|x_i^j)] \text{ and } \tilde{\alpha}_y^j = [\underline{\alpha}_y^j, \overline{\alpha}_y^j] \tag{27}$$

Furthermore, let $\tilde{p}(x_i^j|y) = \tilde{p}(x_i^j|\boldsymbol{\theta}_y^j)$ be expressed as a Gaussian T2 fuzzy set with uncertain mean $\mu$ and fixed standard deviation $\sigma$.

$$\tilde{p}(x_i^j|\boldsymbol{\theta}_y^j) = G(x_i^j|\tilde{\boldsymbol{\theta}}_y^j) = \frac{1}{\sqrt{2\pi(\sigma_y^j)^2}} \exp\left(-\frac{1}{2}\left(\frac{x_i^j - \mu_y^j}{\sigma_y^j}\right)^2\right) \tag{28}$$

which consists of the following upper and lower uncertainty functions,

$$\tilde{p}(x_i^j|\boldsymbol{\theta}_y^j) = [p(x_i^j|\overline{\boldsymbol{\theta}}_y^j), p(x_i^j|\underline{\boldsymbol{\theta}}_y^j)] \tag{29}$$

where

$$p(x_i^j|\overline{\boldsymbol{\theta}}_y^j) = \frac{1}{\sqrt{2\pi(\sigma_y^j)^2}} \exp\left(-\frac{1}{2}\left(\frac{x_i^j - \overline{\mu}_y^j}{\sigma_y^j}\right)^2\right),$$

$$p(x_i^j|\underline{\boldsymbol{\theta}}_y^j) = \frac{1}{\sqrt{2\pi(\sigma_y^j)^2}} \exp\left(-\frac{1}{2}\left(\frac{x_i^j - \underline{\mu}_y^j}{\sigma_y^j}\right)^2\right) \tag{30}$$

After substituting Eqs. (27)-(29) into Eq. (26), the maximization problem becomes

$$\begin{aligned}&\text{Max } \overline{L}_\lambda(\overline{p}(y|\boldsymbol{x}^j), \overline{\boldsymbol{\Theta}}_k^j) = \\ &\frac{1}{N}\sum_{i=1}^{N}\sum_{y=1}^{k^j} \overline{p}(y|x_i^j) \log \overline{\alpha}_y^j G(x_i|\overline{\mu}_y^j, \sigma_y^j) - \\ &\lambda \frac{1}{N}\sum_{i=1}^{N}\sum_{y=1}^{k^j} \overline{p}(y|x_i^j) \log \overline{p}(y|x_i^j), \\ &\underline{L}_\lambda(\underline{p}(y|\boldsymbol{x}^j), \underline{\boldsymbol{\Theta}}_k^j) = \\ &\frac{1}{N}\sum_{i=1}^{N}\sum_{y=1}^{k^j} \underline{p}(y|x_i^j) \log \underline{\alpha}_y^j G(x_i^j|\underline{\mu}_y^j, \sigma_y^j) - \\ &\lambda \frac{1}{N}\sum_{i=1}^{N}\sum_{y=1}^{k^j} \underline{p}(y|x_i^j) \log \underline{p}(y|x_i^j)\end{aligned} \tag{31}$$

s.t.

$$\sum_{y=1}^{k_j} \frac{\overline{p}(y|x_i^j) + \underline{p}(y|x_i^j)}{2} = 1,$$

$$i = 1, 2, \cdots, N, j = 1, 2, \cdots, D \tag{32}$$

This is a constrained multi-objective optimization problem. We use the weighted sum multi-objective optimization technique with combinations of different weightings to solve the problem. Introducing $N$ Lagrange multipliers $\beta_i$ ($i$= 1, 2, $\cdots$, $N$) obtains the following Lagrange function,

$$\begin{aligned}&\text{Max } F = w_1 \overline{L}_\lambda(\overline{p}(y|x_i^j), \overline{\boldsymbol{\Theta}}_k^j) + \\ &w_2 \underline{L}_\lambda(\underline{p}(y|x_i^j), \underline{\boldsymbol{\Theta}}_k^j) + \\ &\sum_{i=1}^{N} \beta_i \left(\sum_{y=1}^{k^j} \frac{\overline{p}(y|x_i^j) + \underline{p}(y|x_i^j)}{2} - 1\right)\end{aligned} \tag{33}$$

where $w_1$ and $w_2$ are the weights of the corresponding individual objective functions and $w_1 + w_2 = 1$ ($w_1 \geqslant 0$, $w_2 \geqslant 0$).

After taking partial derivatives of Eq. (33) with respect to $\overline{p}(y|x_i^j)$, $\underline{p}(y|x_i^j)$, and $\beta_i$, we have $N$ sets of the following equations:

$$\frac{\partial F}{\partial \overline{p}(y|x_i^j)} = w_1 \log \overline{a}_y^j G(x_i^j|\overline{\mu}_y^j, \sigma_y^j) - w_1\lambda(1 + \log \overline{p}(y|x_i^j)) + \frac{\beta_i}{2} = 0 \tag{34}$$

$$\frac{\partial F}{\partial \underline{p}(y|x_i^j)} = w_2 \log \underline{\alpha}_y^j G(x_i^j|\underline{\mu}_y^j, \sigma_y^j) - w_2\lambda(1 + \log \underline{p}(y|x_i^j)) + \frac{\beta_i}{2} = 0 \tag{35}$$

$$\sum_{y=1}^{k^j} \frac{\overline{p}(y|x_i^j) + \underline{p}(y|x_i^j)}{2} = 1 \tag{36}$$

There are two different cases, i.e., $w_1 = w_2$ and $w_1 \neq w_2$, and they are presented below.

(a) $w_1 = w_2$

The detailed mathematical derivation is presented in Appendix A of this paper.

$$\overline{p}(y|x_i^j) = \frac{2(\overline{a}_y^j G(x_i^j|\overline{\mu}_y^j, \sigma_y^j))^{1/\lambda}}{\sum_{y=1}^{k^j}((\overline{a}_y^j G(x_i^j|\overline{\mu}_y^j, \sigma_y^j))^{1/\lambda} + (\underline{\alpha}_y^j G(x_i^j|\underline{\mu}_y^j, \sigma_y^j))^{1/\lambda})} \tag{37}$$

$$\underline{p}(y|x_i^j) = \frac{2(\underline{\alpha}_y^j G(x_i^j|\underline{\mu}_y^j, \sigma_y^j))^{1/\lambda}}{\sum_{y=1}^{k^j}((\overline{a}_y^j G(x_i^j|\overline{\mu}_y^j, \sigma_y^j))^{1/\lambda} + (\underline{\alpha}_y^j G(x_i^j|\underline{\mu}_y^j, \sigma_y^j))^{1/\lambda})} \tag{38}$$

(b) $w_1 \neq w_2$

$$\overline{p}(y|x_i^j) = K_1(\overline{a}_y^j G(x_i^j|\overline{\mu}_y^j, \sigma_y^j))^{1/\lambda},$$
$$\underline{p}(y|x_i^j) = K_2(\underline{\alpha}_y^j G(x_i^j|\underline{\mu}_y^j, \sigma_y^j))^{1/\lambda} \quad (39)$$

where the calculation of $K_1$ and $K_2$ is shown in Appendix B of this paper.

**Step 2**: Fix $\tilde{\boldsymbol{\Theta}}_1$ and solve $\tilde{\boldsymbol{\Theta}}_2$ by maximizing Eq. (25) with respect to $\tilde{\boldsymbol{\Theta}}_2$.

(a) Parameter learning $\tilde{\alpha}_y^j$

The detailed mathematical derivation is presented in Appendix C of this paper.

$$\overline{\alpha}_y^j = 2w_1 \frac{\sum_{i=1}^{N} \overline{p}(y|x_i^j)}{\sum_{y=1}^{k^j} \left\{ w_1 \sum_{i=1}^{N} \overline{p}(y|x_i^j) + w_2 \sum_{i=1}^{N} \underline{p}(y|x_i^j) \right\}} \quad (40)$$

$$\underline{\alpha}_y^j = 2w_2 \frac{\sum_{i=1}^{N} \underline{p}(y|x_i^j)}{\sum_{y=1}^{k^j} \left\{ w_1 \sum_{i=1}^{N} \overline{p}(y|x_i^j) + w_2 \sum_{i=1}^{N} \underline{p}(y|x_i^j) \right\}} \quad (41)$$

(b) Parameter learning $\tilde{\mu}_y^j$

Differentiating Eq. (31) partially with respect to $\overline{\mu}_y^j$ and $\underline{\mu}_y^j$ and equating them to zero yields

$$\frac{\partial \overline{L}_\lambda}{\partial \overline{\mu}_y^j} = \frac{\partial}{\partial \overline{\mu}_y^j} \left\{ \frac{1}{N} \sum_{i=1}^{N} \sum_{y=1}^{k^j} \overline{p}(y|x_i^j) \log \overline{\alpha}_y^j G(x_i^j|\overline{\mu}_y^j, \sigma_y^j) \right\} = 0 \quad (42)$$

$$\frac{\partial \underline{L}_\lambda}{\partial \underline{\mu}_y^j} = \frac{\partial}{\partial \underline{\mu}_y^j} \left\{ \frac{1}{N} \sum_{i=1}^{N} \sum_{y=1}^{k^j} \underline{p}(y|x_i^j) \log \underline{\alpha}_y^j G(x_i^j|\underline{\mu}_y^j, \sigma_y^j) \right\} = 0 \quad (43)$$

After solving Eqs. (42) and (33), we obtain

$$\overline{\mu}_y^j = \frac{\sum_{i=1}^{N} \overline{p}(y|x_i^j) x_i^j}{\sum_{i=1}^{N} \overline{p}(y|x_i^j)} \quad (44)$$

$$\underline{\mu}_y^j = \frac{\sum_{i=1}^{N} \underline{p}(y|x_i^j) x_i^j}{\sum_{i=1}^{N} \underline{p}(y|x_i^j)} \quad (45)$$

(c) Parameter learning $(\sigma_y^2)^j$

Aggregating the multiple objective functions of Eq. (31), we have

$$\text{Max } F = w_1 \overline{L}_\lambda(\overline{p}(y|x_i^j), \overline{\boldsymbol{\Theta}}_k^j) + w_2 \underline{L}_\lambda(\underline{p}(y|x_i^j), \underline{\boldsymbol{\Theta}}_k^j) \quad (46)$$

Taking partial derivatives of Eq. (46) with respect to $(\sigma_y^2)^j$ and equating them to zero yields

$$\frac{\partial F}{\partial (\sigma_y^2)^j} = \frac{w_1}{N} \sum_{i=1}^{N} \overline{p}(y|x_i^j) \left( \frac{(x_i^j - \overline{\mu}_y^j) - 1}{2(\sigma_y^2)^j} \right) + \frac{w_2}{N} \sum_{i=1}^{N} \underline{p}(y|x_i^j) \left( \frac{(x_i^j - \underline{\mu}_y^j) - 1}{2(\sigma_y^2)^j} \right) = 0 \quad (47)$$

After solving Eq. (47), we have

$$(\sigma_y^2)^j = \frac{w_1 \sum_{i=1}^{N} \overline{p}(y|x_i^j)(x_i^j - \overline{\mu}_y^j)^2 + w_2 \sum_{i=1}^{N} \underline{p}(y|x_i^j)(x_i^j - \underline{\mu}_y^j)^2}{w_1 \sum_{i=1}^{N} \overline{p}(y|x_i^j) + w_2 \sum_{i=1}^{N} \underline{p}(y|x_i^j)} \quad (48)$$

## 7 Test Results in Practical Applications

The proposed SDSA-FNN-T2FBYYL is implemented in the MatLab environment. To test and compare the proposed method with other existing relevant methods, we programmed a method based on Neural Networks (NN) and a method based on type-1 FBYYL itself (marked as SDSA-FNN-T1FBYYL). Moreover, to verify the proposed method's adaptability to different load patterns, two scenarios were tested, namely workday and holiday load profiles. After analyzing historical data, load characteristics, and societal activity, the historical hourly system load data could be classified into three data groups representing "Mon", "Tue–Fri", and "Holiday (Sat, Sun & public holidays)" categories. Moreover, given that the main industries in Macao are gambling and hospitality, the load profile on widely celebrated public festivals or holidays such as the Spring Festival, May Day, etc. are similar to those during the weekend (Sat and Sun), as shown in Fig. 3. Therefore, they are grouped in the same category, i.e., holiday.

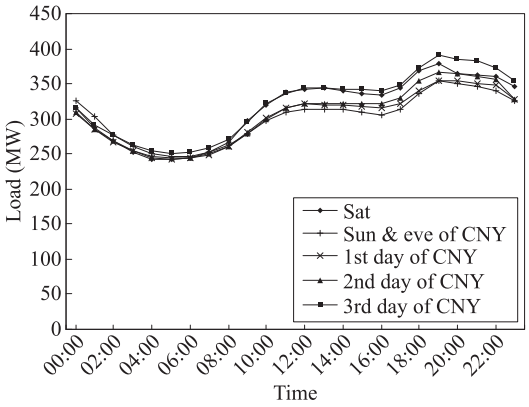


**Fig. 3 Load profiles of public holidays and weekend.**

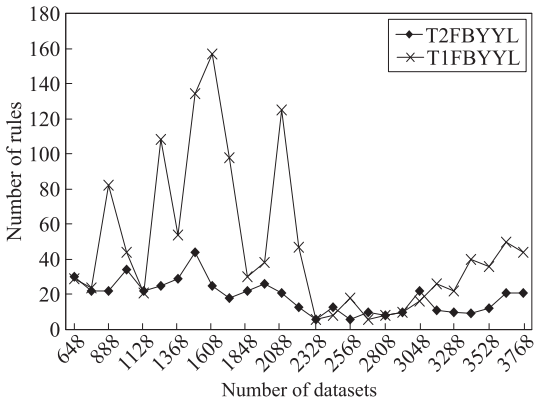


**Fig. 4 Number of generated fuzzy rules with respect to different datasets.**

### 7.1 Test of self-learning ability

A historical operational dataset (from Jan. to Dec., 2010) was collected from Macao electric utility. Macao lies on the western side of Pearl River Delta. The electric load comprises mainly residential and commercial loads.

The SDSA-FNN-T2FBYYL has five input variables and one output variable, as can be inferred from Table 1. First, different datasets are tested, whereby each data set contains different numbers of data entries. Each dataset is tested 10 times by running T1FBYYL and T2FBYYL. The obtained number of fuzzy rules is averaged for performance comparison. As shown in Fig. 4, T2FBYYL generates a more compact system structure in terms of having fewer fuzzy rules. Furthermore, it can be seen that when number of data entries increases, more compact system structures can be obtained. This observation is consistent with one requirement of BYYL, i.e., number of training data should be large enough.

### 7.2 Forecasting 1-day (workday) ahead electric load profile

Next, forecast the 1-day ahead electric load profile. The Mean Absolute Percentage Error (MAPE) performance index is used to evaluate the forecasting accuracy. Data of workday "Tue–Fri" from 1 Feb. to 20 Apr., 2010 were used to train the forecasting system according to our proposed method. Data from 21 Apr. to 30 Apr., 2010 were used for testing.

As shown in Fig. 5, MAPE = 1.687% was obtained for 1-day ahead electric load forecast with the system constructed using SDSA-FNN-T2FBYY. This result is better than that achieved employing the frequently used NN-based forecasting tool, which has MAPE = 1.834%; and better than SDSA-FNN-T1BFYYL, which has MAPE = 1.765% (Table 2). Figure 6 shows the forecast of 1-day ahead electric load profile by SDSA-FNN-T2FBYYL compared with the actual values recorded on 28 Apr., 2010 (Wednesday) by the Supervisory Control And Data Acquisition (SCADA)

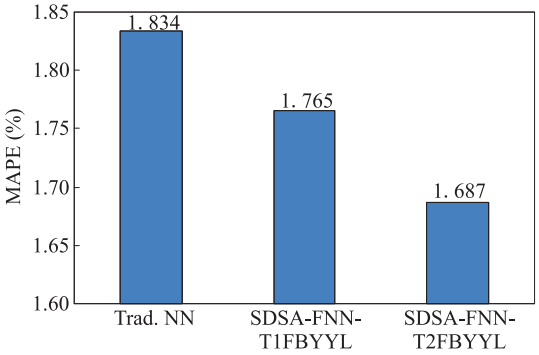


**Fig. 5 Performance comparison of Trad. NN, SDSA-FNN-T1FBYYL, and SDSA-FNN-T2FBYYL (1-day ahead electric load profile (workday)).**

**Table 1 Input and output variables of system.**

| Input variable | Output variable |
|---|---|
| $L_d(t)$: electric load on $d$-th day and $t$-th time interval. | |
| $L_d(t-1)$: electric load on $d$-th day and $(t-1)$-th time interval. | |
| $L_d(t-2)$: electric load on $d$-th day and $(t-2)$-th time interval. | $L_{d+1}(t)$: electric load on $(d+1)$-th day and $t$-th time interval. |
| $L_{d-1}(t)$: electric load on $(d-1)$-th day and $t$-th time interval. | |
| $T_d$: average temperature value on $d$-th day. | |

**Table 2 Number of generated rules and performance comparison (MAPE).**

| Load profile | Algorithm name | Number of rules/neurons | Number of FNN parameters | MAPE (%) |
|---|---|---|---|---|
| Workday | Trad. NN | 20# | — | 1.834 |
| | SDSA-FNN-T1FBYYL | 36* | 110 | 1.765 |
| | SDSA-FNN-T2FBYYL | 16* | 50 | 1.687 |
| Holiday | Trad. NN | 20# | — | 2.585 |
| | SDSA-FNN-T1FBYYL | 18* | 56 | 2.781 |
| | SDSA-FNN-T2FBYYL | 16* | 50 | 1.921 |

*: fuzzy rules, #: neurons

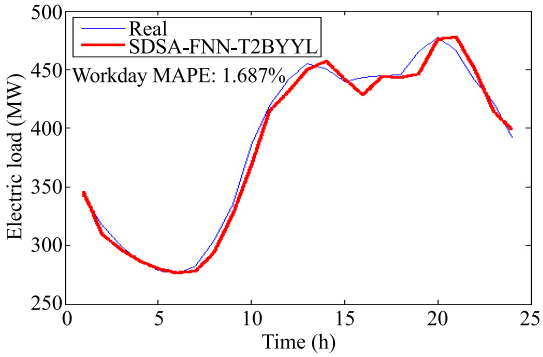


**Fig. 6 Forecast of 1-day (workday) ahead electric load profile and corresponding actual values (Real: actual electric load values; SDSA-FNN-T2FBYYL: forecast electric load values).**

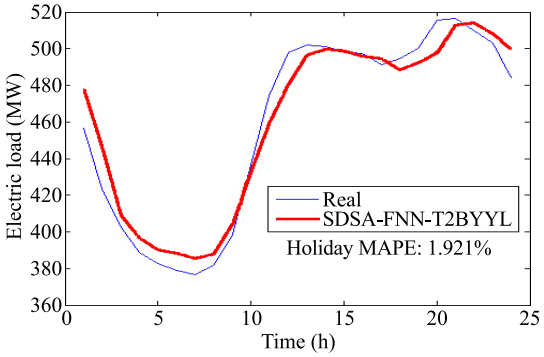


**Fig. 7 Forecast of 1-day (holiday) ahead electric load profile and corresponding actual values (Real: actual electric load values; SDSA-FNN-T2FBYYL: forecast electric load values).**

system installed at the dispatch center of Macao electric utility.

### 7.3 Forecast of 1-day (holiday) ahead electric load profile

Since electric load profiles during holiday (Saturday, Sunday, and public holiday) are different from the ones during workday, SDSA-FNN-T2FBYYL is tested further using the holiday profile. For the test, data of holiday profiles (Sat, Sun, and Holiday) from 1 Feb. to 19 Jun., 2010, and data from 20 Jun. to 30 Jun., 2010, were used for training and testing, respectively.

Figure 7 shows the forecast of 1-day ahead electric load profile, which agrees well with the actual values recorded on 27 Jun., 2010, (Sunday) with acceptable accuracy.

### 7.4 Forecasting 1-month electric load profile

Finally, SDSA-FNN-T2FBYYL is further tested to carry out 1-month forecast. Figure 8 shows the performance when forecasting 1-month electric loads dated 12 Jun. to 9 Jul., 2010.

The above testing results reveal the following main advantages of our proposed methodology:

(1) By integrating the T2 fuzzy theory with BYYL, a more compact system structure can be generated.

(2) If a larger dataset is used, a more compact system structure can be sought out by our approach. In addition, the observation is consistent with one requirement of BYYL. However, the trade-off between system compactness and computing cost should be well balanced.

(3) As shown in Figs. 5, 8, and 9, SDSA-FNN-T2FBYYL outperforms the other relevant methods because it can handle data/information uncertainties that the other methods cannot.

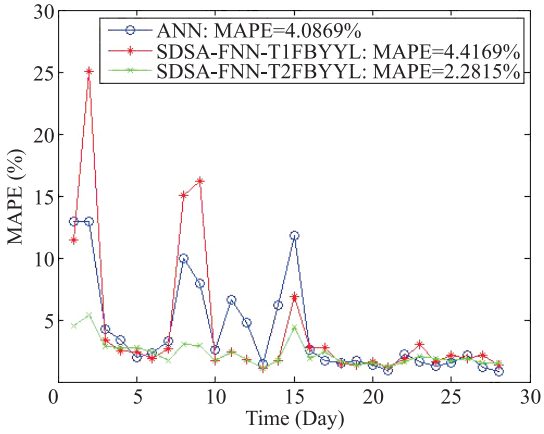


**Fig. 8 Performance when forecasting 1-month electric load profiles (1-month load forecasting performance).**

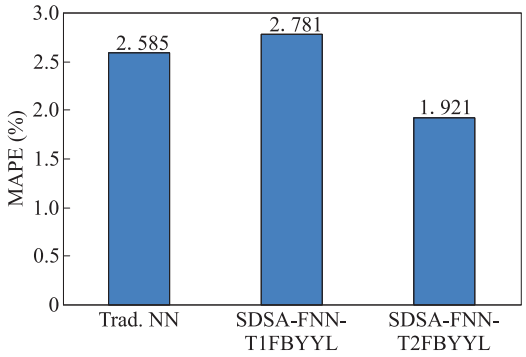


**Fig. 9 Performance comparison of Trad. NN, SDSA-FNN-T1FBYYL, and SDSA-FNN-T2FBYYL (1-day ahead electric load profile (holiday)).**

(4) It is known that if fewer fuzzy rules are used, system forecasting accuracy might decrease and vice versa. However, as shown in the application examples, our proposed method can not only construct the best structure automatically but also achieve a high-performance system simultaneously.

(5) The proposed method SDSA-FNN-T2FBYYL is promising for application to electric load forecasting, which is characterized as being stochastic and uncertainty prone, as shown in the various tests described in this study.

## 8 Conclusions and Outlook

In the age of smart grid and Internet of Things, electric load forecasting techniques are required to have smart abilities and be highly capable of tackling various uncertainties. The proposed integration of the T2 fuzzy theory with BYYL makes a system to automatically construct the best system structure, while ensuring forecasting accuracy and enhanced capability of tackling uncertainties.

Moreover, the hybridization/integration creates a multi-objective optimization problem. The weighted sum multi-objective optimization technique with combinations of different weightings is implemented to achieve the best trade-off among the multiple objectives of T2 fuzzy BYYL.

The validation of the proposed methodology using a real dataset collected from Macao electric utility shows promising features and profound advantages from the viewpoint of electric load forecasting.

Future study would involve research and application of other novel multi-objective optimization techniques or promising evolutionary algorithms to solve the multi-objective T2 fuzzy BYYL optimization problem.

## Acknowledgements

This study was supported by the Research Committee of University of Macau with Grant No. MYRG2014-00060-FST and the Science and Technology Development Fund (FDCT) of Macau S.A.R with Grant No. 016/2012/A1.

## References

[1] M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper-Saddle River, NJ, USA: Prentice-Hall, 2001.

[2] Q. L. Liang and J. M. Mendel, Interval type-2 fuzzy logic system: Theory and design, *IEEE Trans. on Fuzzy Systems*, vol. 8, no. 5, pp. 535–550, 2000.

[3] M. Mendel and R. I. John, Type-2 fuzzy sets made simple, *IEEE Trans. on Fuzzy Systems*, vol. 10, no. 2, pp. 117–127, 2002.

[4] M. Mendel, R. I. John, and F. L. Liu, Interval type-2 fuzzy logic systems made simple, *IEEE Trans. on Fuzzy Systems*, vol. 14, no. 6, pp. 808–821, 2006.

[5] C. F. Juang and Y. W. Tsao, A self-evolving interval type-2 fuzzy neural network with online structure and parameter learning, *IEEE Trans. on Fuzzy Systems*, vol. 16, no. 6, pp. 1411–1424, 2008.

[6] Z. Q. Liu and Y. K. Liu, Type-2 fuzzy variables and their arithmetic, *Soft Computing*, vol. 14, pp. 729–747, 2010.

[7] G. C. Liao and T. P. Tsao, Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting, *IEEE Trans. Evolutionary Computing*, vol. 10, no. 3, pp. 330–340, 2006.

[8] Z. Yun, Z. Quan, S. C. Xin, L. S. Lan, L. Y. Ming, and S. Yang, RBF neural network and ANFIS-based short-term load forecasting approach in real-time price environment, *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 853–858, 2008.

[9] M. Hanmandlu and B. K. Chauhan, Load forecasting using hybrid models, *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 20–29, 2011.

[10] L. Xu, Bayesian Ying-Yang machine, clustering and number of clusters, *Pattern Recognition Letters*, vol. 18, pp. 1167–1178, 1997.

[11] L. Xu, RBF nets, mixture experts, and Bayesian Ying-Yang learning, *Neurocomputing,* vol. 1, pp. 223–257, 1998.

[12] L. Xu, Best harmony, unified RPCL and automated model selection for unsupervised and supervised learning on Gaussian mixtures, three-layer nets and ME-RBF-SVM models, *International Journal of Neural Systems*, vol. 11, no. 1, pp. 43–69, 2001.

[13] L. Xu, BYY harmony learning, independent state space, and generalized APT financial analysis, *IEEE Trans. on Neural Networks*, vol. 12, no. 4, pp. 822–849, 2001.

[14] L. Xu, BYY harmony learning, structural RPCL, and topological self-organizing on mixture models, *Neural Networks*, vol. 15, pp. 1125–1151, 2002.

[15] L. Xu, *Ying-Yang Learning—The Handbook of Brain Theory and Neural Networks*, 2nd ed. The MIT Press, 2002, pp. 1231–1237.

[16] J. W. Ma, T. J. Wang, and L. Xu, An annealing approach to BYY harmony learning on Gaussian mixture with automated model selection, presented in IEEE Int. Conf. Neural Networks & Signal Processing, Nanjing, China, 2003, pp. 23–28.
[17] J. W. Ma and J. F. Liu, The BYY annealing learning algorithm for Gaussian mixture with automated model selection, *Pattern Recognition*, vol. 40, pp. 2029–2037, 2007.
[18] J. W. Ma and X. F. He, A fast fixed-point BYY harmony learning algorithm on Gaussian mixture with automated model selection, *Pattern Recognition Letters*, vol. 29, pp. 701–711, 2008.
[19] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Syst., Man, Cybern.*, vol. 15, no. 1, pp. 116–132, 1985.
[20] L. Wang, Fuzzy systems are universal approximators, in *Proc. 1st IEEE Conf. Fuzzy Syst.*, San Diego, CA, USA, 1992, pp. 1163–1169.
[21] J. S. R. Jang, ANFIS: Adaptive-network-based fuzzy inference system, *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 665–684, 1993.
[22] C. F. Juang and C. T. Lin, An on-line self-constructing neural fuzzy inference network and its applications, *IEEE Trans. on Fuzzy Systems*, vol. 6, no. 1, pp. 12–32, 1998.
[23] E. D. Lughofer, FLEXFIS—A robust incremental learning approach for evolving Takagi-Sugeno fuzzy models, *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 6, pp. 1393–1409, 2008.
[24] D. Wang, X. J. Zeng, and J. A. Keane, An evolving-construction scheme for fuzzy systems, *IEEE Trans. on Fuzzy Systems,* vol. 18, no. 4, pp. 755–770, 2010.
[25] S. L. Chiu, Fuzzy model identification based on cluster estimation, *J. Intell. Fuzzy Syst.*, vol. 2, pp. 267–278, 1994.
[26] L. Wang and R. Langari, Building Sugeno-type models using fuzzy discretization and orthogonal parameter estimation techniques, *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 4, pp. 454–458, 1995.
[27] Y. Shi, M. Mizumoto, and P. Shi, Fuzzy if-then rule generation based on neural network and clustering algorithm techniques, in *Proceedings of IEEE Tencon'02*, 2002.

## Appendix A: System parameter learning by multi-objective optimization

Solving Eqs. (34)-(36) with $w_1 = w_2$, we obtain

$$\log(\overline{a}_y^j G(x_i^j|\overline{\mu}_y^j,\ \sigma_y^j))^{1/\lambda} - 1 - \log \overline{p}(y|x_i^j) + \frac{\beta_i}{2w_1\lambda} = 0,$$
$$\log(\underline{\alpha}_y^j G(x_i^j|\underline{\mu}_y^j, \sigma_y^j))^{1/\lambda} - 1 - \log \underline{p}(y|x_i^j) + \frac{\beta_i}{2w_2\lambda} = 0 \quad \text{(A1)}$$

Let $\log K = \left(\frac{\beta_i}{2w_1\lambda} - 1\right) = \left(\frac{\beta_i}{2w_2\lambda} - 1\right)$, then we have

$$\overline{p}(y|x_i^j) = K(\overline{a}_y^j G(x_i^j|\overline{\mu}_y^j, \sigma_y^j))^{1/\lambda},$$
$$\underline{p}(y|x_i^j) = K(\underline{\alpha}_y^j G(x_i^j|\underline{\mu}_y^j, \sigma_y^j))^{1/\lambda} \quad \text{(A2)}$$

After summing $k^j$ sets of Eq. (A2), we obtain

$$K = \frac{2}{\sum_{y=1}^{k^j} ((\overline{a}_y^j G(x_i^j|\overline{\mu}_y^j, \sigma_y^j))^{1/\lambda} + (\underline{\alpha}_y^j G(x_i^j|\underline{\mu}_y^j, \sigma_y^j))^{1/\lambda})} \quad \text{(A3)}$$

Substituting the parameter $K$ into Eqs. (A2), we have Eqs. (37) and (38) in the main body of the paper.

## Appendix B: System parameter learning by multi-objective optimization

Let $\frac{\beta_i}{2w_1\lambda} - 1 = \log K_1$ and $\frac{\beta_i}{2w_2\lambda} - 1 = K_2$, then Eqs. (34) and (35) become

$$\log(\overline{a}_y^j G(x_i^j|\overline{\mu}_y^j,\ \sigma_y^j))^{1/\lambda} - \log \overline{p}(y|x_i^j) + \log K_1 = 0,$$
$$\log(\underline{\alpha}_y^j G(x_i^j|\underline{\mu}_y^j, \sigma_y^j))^{1/\lambda} - \log \underline{p}(y|x_i^j) + \log K_2 = 0 \quad \text{(B1)}$$

Equation (B1) yields

$$\overline{p}(y|x_i^j) = K_1(\overline{a}_y^j G(x_i^j|\overline{\mu}_y^j, \sigma_y^j))^{1/\lambda},$$
$$\underline{p}(y|x_i^j) = K_2(\underline{\alpha}_y^j G(x_i^j|\underline{\mu}_y^j, \sigma_y^j))^{1/\lambda} \quad \text{(B2)}$$

After summing $k^j$ sets of Eq. (B2) and noting that $\sum_{y=1}^{k^j} \frac{\overline{p}(y|x_i^j) + \underline{p}(y|x_i^j)}{2} = 1$, we obtain

$$\sum_{y=1}^{k^j} (K_1(\overline{a}_y^j G(x_i^j|\overline{\mu}_y^j, \sigma_y^j))^{1/\lambda} + K_2(\underline{\alpha}_y^j G(x_i^j|\underline{\mu}_y^j, \sigma_y^j))^{1/\lambda}) = 2 \quad \text{(B3)}$$

Since

$$w_1 \log K_1 + w_1 = w_2 \log K_2 + w_2 \quad \text{(B4)}$$

Solving Eqs. (B3) and (B4) together yields

$$w_1 \log K_1 - w_2 \log\left(\frac{2 - K_1 A}{B_2}\right) + w_1 - w_2 = 0 \quad \text{(B5)}$$

where

$$A = \sum_{y=1}^{k^j} ((\overline{a}_y^j G(x_i^j|\overline{\mu}_y^j, \sigma_y^j))^{1/\lambda}),$$
$$B = \sum_{y=1}^{k^j} ((\underline{\alpha}_y^j G(x_i^j|\underline{\mu}_y^j, \sigma_y^j))^{1/\lambda}).$$

Finally, we can obtain $K_1$ and $K_2$.

## Appendix C: System parameter learning by multi-objective optimization

Introducing another Lagrange multiplier $\beta$ in Eq. (31), we have

$$\text{Max } F = w_1 \overline{L}_\lambda(\overline{p}(y|\boldsymbol{x}^j), \overline{\boldsymbol{\Theta}}_k^j) +$$

$$w_2 \underline{L}_\lambda(\underline{p}(y|\boldsymbol{x}^j), \underline{\boldsymbol{\Theta}}_k^j) + \beta\left(\sum_{y=1}^{k^j} \frac{\overline{\alpha}_y^j + \underline{\alpha}_y^j}{2} - 1\right) \quad \text{(C1)}$$

Taking partial derivatives of Eq. (C1) with respect to $\overline{\alpha}_y^j$, $\underline{\alpha}_y^j$, and $\beta$ and setting them to zero yields the following series of equations.

$$\frac{\partial F}{\partial \overline{\alpha}_y^j} = \frac{w_1}{N}\sum_{i=1}^{N} \frac{\overline{p}(y|x_i^j)}{\overline{\alpha}_y^j} - \frac{\beta}{2} = 0 \quad \text{(C2)}$$

$$\frac{\partial F}{\partial \underline{\alpha}_y^j} = \frac{w_2}{N}\sum_{i=1}^{N} \frac{\underline{p}(y|x_i^j)}{\underline{\alpha}_y^j} - \frac{\beta}{2} = 0 \quad \text{(C3)}$$

$$\frac{\partial F}{\partial \beta} = \sum_{y=1}^{k^j} \frac{\overline{\alpha}_y^j + \underline{\alpha}_y^j}{2} - 1 = 0 \quad \text{(C4)}$$

Solving this series, we obtain

$$\beta\overline{\alpha}_y^j = \frac{2w_1}{N}\sum_{i=1}^{N} \overline{p}(y|x_i^j) \quad \text{(C5)}$$

$$\beta\underline{\alpha}_y^j = \frac{2w_2}{N}\sum_{i=1}^{N} \underline{p}(y|x_i^j) \quad \text{(C6)}$$

Then, Eq. (C7) is obtained

$$\beta = \frac{\sum_{y=1}^{k^j}\left\{\frac{w_1}{N}\sum_{i=1}^{N}\overline{p}(y|x_i^j) + \frac{w_2}{N}\sum_{i=1}^{N}\underline{p}(y|x_i^j)\right\}}{\sum_{y=1}^{k^j}\frac{\overline{\alpha}_y^j + \underline{\alpha}_y^j}{2}} \quad \text{(C7)}$$

Substituting Eq. (C7) into Eqs. (C5) and (C6), we can obtain Eqs.(40) and (41).

**Mingchui Dong** is currently a professor and PhD supervisor at Department of Electrical and Computer Engineering of University of Macau, Automation Department of Tsinghua University of China and professor of YanTai University of China. He received his MS degree in EEE in 1975 from Tsinghua University. He was a visiting scholar in 1981 at Rome University, Italy.

**Chinwang Lou** received his bachelor degree and master degree of electrical and electronic engineering in 1994 and 2006 from University of Macau. He is currently pursuing towards PhD degree in Electrical and Computer Engineering Department at University of Macau. His current research is oriented to R&D hybrid intelligent and adaptive systems applicable to electric load forecasting.