CIRJE-F-1067

State Space Approach to Adaptive Fuzzy Modeling: Application to Financial Investment

Masafumi Nakano
Graduate School of Economics, The University of Tokyo
Akihiko Takahashi
The University of Tokyo
Soichiro Takahashi
Graduate School of Economics, The University of Tokyo

October 2017

State Space Approach to Adaptive Fuzzy Modeling: Application to Financial Investment

Masafumi Nakano, Akihiko Takahashi, and Soichiro Takahashi

***Abstract*—This paper proposes a new state space approach to adaptive fuzzy modeling under the dynamic environment, where Bayesian filtering sequentially learns the model parameters including model structures themselves as state variables. In particular, our approach specifies the state transitions as mean-reversion processes, which intends to incorporate and extend the established state-of-art learning techniques as follows: First, the mean-reversion levels of model parameters are determined by applying some existing learning method to a training period. Next, filtering implementation over test data enables on-line estimation of the parameters, where the estimates are adaptively tuned for each new data arrival based on the obtained reliable learning result. In this work, we concretely design a Takagi-Sugeno-Kang fuzzy model for financial investment, whose parameters follow autoregressive processes with the mean-reversion levels decided by particle swarm optimization. Since there exist Monte Carlo simulation-based algorithms called particle filtering, our methodology is applicable to a quite general setting including non-linearity, which actually arises in our investment problem. Then, an out-of-sample numerical experiment with security price data successfully demonstrates its effectiveness.**

***Index Terms*—fuzzy system, adaptive learning, state space model, particle filtering, financial investment.**

I. Introduction

RECENTLY, artificial intelligence (AI) has been increasingly important in both academia and industry, which is supported by development of computer performance including software packages, as well as accessibility to wide-rage and large-scale of data.

In general, machine learning is an inevitable step to attain the objective of AI, that is, making intelligent machine or computer. Especially from a probabilistic perspective, a task of machine learning can be interpreted as acquiring latent characteristics or patterns from observed data under the existence of uncertainty related to data measurement, parameter estimation and model specification [1].

Historically, many researchers have developed various effective learning algorithms for different AI models. For instance, (deep) learning to artificial neural networks (ANNs) is one of the most thriving research topics, such as back propagation algorithm with stochastic gradient decent methods [2], [3]. Moreover, evolutionary calculations such as genetic algorithm (GA) [4], [5] or particle swarm optimization (PSO) [6] are population-based optimization algorithms searching efficiently in complex target spaces.

This research is supported by CARF (Center for Advanced Research in Finance). Also, this work is supported by JSPS KAKENHI Grant Numbers JP17J09046 and JP17J09127.

The authors are with Graduate School of Economics, the University of Tokyo, Tokyo, Japan.

In this paper, we focus on the learning of fuzzy systems. Fuzzy logic [7] is a powerful AI technique for modeling human imprecise perception, knowledge and reasoning. Specifically, a fuzzy system stores multiple IF-THEN inference rules with linguistic expressions called fuzzy sets, which enables to quantitatively imitate human reasoning. Therefore, fuzzy model identification including its parameter learning is thought to be a considerably important issue, which has led to various previous researches.

For instance, one of the most famous approaches is network-based fuzzy modeling [8], [9], where the learning algorithms of ANNs such as back propagation are utilized for the identification by interpreting a fuzzy system as a multi-layered ANN. In addition, there are also a great deal of studies to use evolutionary calculation methods. GA is well-known to the optimal rule generation and parameter learning [10], [11], which is named as genetic fuzzy systems (GFSs) in [12]. Besides, other evolutionary calculation methods such as PSO are also effectively utilized [13]. Moreover, a recent research [14] exploits Markov Chain Monte Carlo (MCMC) method to the estimation of membership functions' parameters for a Mamdani fuzzy model [15] with an application to data in the financial services industry.

Although those established approaches are now widely accepted with great success, there is a room for improvement particularly in the learning problems under the drastically changing environment, which are often observed as time-series data with limited stationarity. Unfortunately, since the non-stationary environmental change is also easy to cause over-fitting, it is a challenging task to develop high-performance fuzzy models under this environment.

One of the most reasonable approaches to this problem is to sequentially update model structure and parameters along with the dynamic changes of the environment. Notably, evolving fuzzy [16]–[20] and neuro fuzzy [21]–[24] systems take this approach, in which a rule base and parameters continually evolve to match the current information from new data. Especially, participatory evolving fuzzy systems [25] are applied to modeling financial time-series in [26]–[28].

In the current work, we apply a state space model with Bayesian filtering algorithms to adaptive fuzzy modeling. State space models are commonly used in various fields to represent dynamic dependence between latent state and observed variables. Their dynamics are described as time-series models called state and observation equations. Moreover, Bayesian filtering algorithms make it possible to estimate the time transitions of unobservable state variables.

Particularly, in our state space framework, the parameters of fuzzy systems are interpreted as unobservable state variables,

which are sequentially learned for each new data measurement by Bayesian filters. Note that by using Monte Carlo simulation-based filtering algorithms called particle filters (PFs), our scheme is applicable to a wide range of problem settings including non-linear and non-Gaussian properties, as well as various types of fuzzy reasoning methods such as Mamdani or high order TSK models. Further, it is remarkable that our learning scheme can include the model selection. More precisely, our state space representation allows the model structures themselves to be added to the state variables, which possibly shows a new direction of Bayesian dynamic model selection.

The main contribution of this work is to actively introduce time-series models into the parameter transitions of fuzzy systems, which are sequentially estimated within a state space framework with PF algorithm. Thereby, one of the most promising applications of our scheme is to learn model parameters around specific levels. That is, the following procedure enables systematic generation of a new adaptive learning scheme based on an existing state-of-art method: First, the existing learning algorithm is applied to training data, which gives us parameter estimates of a fuzzy system. Then, by assuming that the state variables follow stochastic processes moving around the obtained estimates, filtering implementation over test data makes it possible to continuously adapt model parameters to the dynamic environment based on the existing method. Importantly, this formulation is generally expected to enhance the performances of the established reliable approaches mentioned above [8]–[14].

In this study, after introducing a zero order Takagi-Sugeno-Kang (TSK) fuzzy model [29], [30] for financial investment, we provide a state space representation of its parameter learning problem. Here, the parameter transitions, i.e. state equations, are specified as autoregressive processes of order one (AR(1) processes), which have tendencies to drift toward the long-term mean levels. In a numerical experiment with actual financial security price data, we execute PSO for a training period to decide these long-term levels. Then, over test data, PF sequentially adapts the fuzzy model parameters to changing environment in real time, which is expected to enhance the reliable PSO learning results. In other words, we present a new adaptive fuzzy model with PSO by exploiting our methodology.

The remainder of this paper is organized as follows. Section II explains our methodology of machine learning with state space framework. After introducing our fuzzy system whose learning problem is concretely formulated in our state space representation, Section III shows an out-of-sample numerical experiments. Section IV refers to the applicability of our approach to dynamic model selection. Then, Section V concludes. Appendix summarizes a detailed algorithm of PF used in this work.

II. Methodology

A. State Space Model

State space framework assumes that there exist unobservable latent/state variables driven by stochastic processes behind observed time-series data. Precisely, it consists of the following observation and state equations.

$$\begin{aligned} Y_t &= H(Z_t, u_t), \quad \text{[observation equation]} \\ Z_t &= F(Z_{t-1}, v_t), \quad \text{[state equation]} \end{aligned} \tag{1}$$

where Y_t and Z_t are a vector of observable and latent variables, respectively. Besides, u_t and v_t denote random variables for observation and state transition, respectively. Here, the functions H and F are non-linear functions.

In general, Bayesian filtering technique is an on-line statistical approach to sequentially estimate the latent variables Z_t through the observed data $Y_{1:t} \equiv (Y_1, \cdots, Y_t)$ until time t, that is, to estimate the distribution $p(Z_t|Y_{1:t})$. Since updating the estimation for each time of new data arrival, the filtering approach is appropriate for reflecting dynamic changes of environment.

Particularly, particle filtering (PF) resorts to Monte Carlo simulation for latent variables' estimation, whereby it is applicable to non-linear and non-Gaussian settings. This generality is essential to apply PF to machine learning under various situations in practice. The detailed algorithm of our PF is described in Appendix.

B. Machine Learning with State Space Framework

First, let us introduce our methodology in a relatively general setting of supervised machine learning, which will be focused on fuzzy systems in the next Section III. Most of the artificial intelligence (AI) models, such as linear regression/classification models, artificial neural networks, fuzzy systems and so on, are represented as non-linear functions as follows.

$$O_t = f(I_t; Z_t), \tag{2}$$

where f represents an AI system, that is a non-linear mapping from inputs I_t to outputs O_t for given parameters Z_t. Here, since this study discusses machine learning for time-series data, we assign the time index $t \in \{1, \cdots, T\}$ to the variables (I_t, O_t, Z_t). Note that in the standard machine learning problems, the parameter vector Z_t does not have the index t, because it basically intends to decide fixed values or ranges for model parameters from the pairs of input-output data $(I_t, O_t)_t$.

However, especially for machine learning under the dynamic environment such as financial markets, it is also effective to update the model parameter estimates over time, because the optimal parameters (or models) may largely change along with the non-stationary shift of the environment.

Therefore, this study assumes that model parameters are time-varying variables denoted by Z_t, which also helps to mitigate over-fitting problems. Let us remark that this learning problem of the parameter vector Z_t can include the selection of AI models themselves. In other words, Z_t may stand for the AI model structure performing well under the current environment, which possibly shows a new direction of Bayesian dynamic model selection, as discussed in Section IV.

In this work, our learning problem is formulated by the following state space representation.

$$Y_t = H(O_t, u_t), \quad (3)$$
$$O_t = f(I_t; Z_t), \quad (4)$$
$$Z_t = F(Z_{t-1}, v_t). \quad (5)$$

Let us remember that Eq. (3) and (5) denote the observation and state equations, respectively. Also, we note that the AI model Eq. (4) can be included in the both equations.

Behind this formulation, the next situation is assumed: First of all, a designer of AI models under the dynamic environment would like to know which modeling and/or parameters will perform the best, though it is unknown in advance. Then, based on the assumption that the optimal model and/or parameters change over time, the designer attempts to estimate their transitions through periodical performance measurements.

This situation is described by the state space representation in Eq. (3)-(5) as follows: First, the dynamic transitions of optimal model structure and parameters are formulated by the state equation (5). Then, the AI model of Eq. (4) is applied to the input variables I_t for the model parameters Z_t given by the state equation (5), which generates the output variables O_t. Moreover, the observation equation (3) expresses performance measurement with some uncertainty or noise. (Sometimes extremely high performance arises just by a luck, for instance.) Also, it is notable that the performance evaluation depends on the choice of measures.

Based on the above state space representation, our PF algorithm is implemented as follows: First, it generates various candidates, which are referred as "particles" in Appendix, of model's structure and/or parameters according to the state equation, and then selects well-performing ones from the measurement by the observation equation. Let us remark that PF, which allows non-linear and non-Gaussian settings, makes it possible to apply our approach to various kinds of learning problems.

Hereafter, we explain our approach more concretely by showing its application to fuzzy systems.

III. State Space Approach to Fuzzy Modeling

In this section, we apply the methodology described in Section II to learning fuzzy systems.

A. Fuzzy System

A fuzzy system is a non-linear real-valued function, which expresses human imprecise reasoning with a collection of fuzzy IF-THEN rules, that is a fuzzy rule base. To implement the fuzzy rule-based inference, fuzzifier and defuzzifier are applied as interfaces to inputs and outputs of the system, respectively.

One of the most famous fuzzy inference methods is proposed in [29], [30], the so-called Takagi-Sugeno-Kang (TSK) model, whose IF-THEN rule takes the following form:

IF (x_1 is A_1) AND $\cdots$ AND (x_I is A_I),
THEN $y = c(x_1, \cdots, x_I)$.

Here, $(x_1, \cdots, x_I)$ of the premise parts (IF parts) denote input variables which are fuzzified by fuzzy sets $A_1, \cdots, A_I$ with membership functions $\mu_{A_1}, \cdots, \mu_{A_I}$. On the other hand, the output y of the TSK inference is represented by a real-valued function $c(x_1, \cdots, x_I)$, where the case of a linear function called a first-order TSK model is the most well-known. Note that although the premise propositions are connected with AND-operators, there are other choices, e.g. "OR".

In this study, we employ a zero-order TSK type of fuzzy systems, i.e. the case of $c(x_1, \cdots, x_I) = c \in \mathbb{R}$. Here, we suppose there are the following K number of IF-THEN rules:

IF (x_1 is $A_{1,k}$) AND $\cdots$ AND (x_I is $A_{I,k}$),
THEN $y = c_k$,

where $c_k \in \mathbb{R},\ k = 1, \cdots, K$. Then, its typical procedure is described as follows:

(i) [Fuzzification] Input crisp variables $x := (x_1, \cdots, x_I)$ are fuzzified with fuzzy sets $A_{i,k},\ i = 1, \cdots, I,\ k = 1, \cdots, K$, where fuzzy sets $A_{i,k}$ are characterized by their membership functions $\mu_{A_{i,k}} : \mathbb{R} \to [0, 1]$.

(ii) [Fuzzy inference] For $k = 1, \cdots, K$, the firing strength $M_k(x)$ of the k-th IF-THEN rule is calculated as follows.

$$M_k(x) = \prod_{i=1}^{I} \mu_{A_{i,k}}(x_i), \quad (6)$$

Here, fuzzy AND operation is identified with product.

(iii) [Defuzzification] The final output y is calculated as a weighted average of the outputs $(c_k)_{k=1,\cdots,K}$ for all the IF-THEN rules.

$$y = \sum_{k=1}^{K} \frac{M_k(x) c_k}{\sum_{k'=1}^{K} M_{k'}(x)} \quad (7)$$

This defuzzifier is often referred as a centroid method.

Therefore, this fuzzy system is expressed as a non-linear function $\mathbb{R}^I \to \mathbb{R}$. Let us remark that although this study deals with a zero-order TSK type of fuzzy systems, our approach is applicable to other types such as higher order TSK or Mamdani models [15]. As well, the premise parts can include the other operations such as "OR".

Now, we discuss the learning problem of the fuzzy systems. That is, one of the most important issues with fuzzy system applications is how to decide the parameters. Especially, the parameters in output parts of zero-order TSK models, i.e. $(c_k)_k$ in the above example, have critical impact not only on the performance but also on the characteristics of inference. For instance, if c_k is set to be zero for a particular rule, this rule is excluded from the rule base in effect. Moreover, it is clear that a sign of c_k, i.e. positive or negative, may lead the opposite inference result. In that sense, with regard to zero-order TSK models, this parameter estimation is essentially the same as the exploration of the optimal rule base.

As stated in Section I, there have been various established state-of-art techniques to the learning problem of fuzzy systems. Those existing approaches mainly decide fixed model parameter values by using all the given training data. On the other hand, our Bayesian filtering-based methodology sequentially adjusts the estimates of parameters for each new

data arrival. In the subsequent sections, we demonstrate its effectiveness by developing a fuzzy system for asset allocation problems.

B. Fuzzy System for Asset Allocation Problem

This section discusses an asset allocation problem for stocks and bonds using a fuzzy system. Specifically, we consider the situation that an investor executes the trading at each end of month under a no-short sale constraint and no leverage condition, whose restrictions strictly preserve the comparability to benchmark traditional investment strategies.

As for the stock and bond allocation, holding them at a constant ratio is a traditionally popular investment policy in practice. However, it also seems reasonable to change the allocation depending on market conditions. Then, we construct a fuzzy system, which decides allocations (i.e. portfolio weights) based on the information of past time-series data of security returns. Here, the inputs of our fuzzy system are expected returns, volatilities and correlations of trading securities, which are calculated by exponential moving average (EMA) methods, whereas the outputs are rating points of the securities. Let us remark that EMA methods are often used both in practice and academia [31]–[34] due to its high tractability.

TABLE I
IF-THEN RULES OF FUZZY SYSTEM

IF			THEN
Ret.	Vol.	Corr.	security rating
high	*high*	*high*	c_1
high	*high*	*low*	c_2
high	*low*	*high*	c_3
high	*low*	*low*	c_4
low	*high*	*high*	c_5
low	*high*	*low*	c_6
low	*low*	*high*	c_7
low	*low*	*low*	c_8

Table I summarizes the fuzzy IF-THEN rule base employed in this work, where *high* and *low* are fuzzy sets. For instance, the IF-THEN rule corresponding to the 1st row of Table I is as follows.

> IF (return is *high*) AND (volatility is *high*) AND (correlation is *high*), THEN a rating point of the security is $c_1 \in [0,1]$,

where we suppose the parameter $c_k \in [0,1]$ for normalization ($k = 1, \cdots, 8$). Note that correlation of the j-th security indicate those between the 1st and j-th ones ($j = 1, \cdots, N$). Importantly, we assume that the securities' rating points $(c_k)_{k=1,\cdots,8}$ of the fuzzy IF-THEN rules are time-varying $(c_{k,t})_{k=1,\cdots,8}$, which will be sequentially learned in our state-space framework as state variables.

With regard to the membership functions of the fuzzy sets, *high* and *low*, we use the following triangular ones, respectively.

$$\begin{aligned}\mu_{h,i}(x_{i,j,t}) &= \min\left\{1, \max\left\{\frac{x_{i,j,t}-b_{i,t}}{a_{i,t}-b_{i,t}}, 0\right\}\right\},\\ \mu_{l,i}(x_{i,j,t}) &= \min\left\{1, \max\left\{-\frac{x_{i,j,t}-a_{i,t}}{a_{i,t}-b_{i,t}}, 0\right\}\right\},\end{aligned} \tag{8}$$

where $x_{i,j,t}$ denotes the i-th input variable ($i = 1, 2, 3$, i.e. expected return, volatility or correlation) of the j-th security at time t. As stated above, we use the next EMAs for these inputs $(x_{i,j,t})_{i,j,t}$:

$$\begin{aligned}x_{1,j,t} &= \alpha r_{j,t-1} + (1-\alpha)x_{1,j,t-1},\\ x_{2,j,t} &= \sigma_{j,j,t},\\ x_{3,j,t} &= \sigma^2_{1,j,t}/(\sigma_{j,j,t}\sigma_{1,1,t}),\end{aligned} \tag{9}$$

where

$$\begin{aligned}\sigma^2_{j,j',t} &= (1-\alpha)\sigma^2_{j,j',t-1}\\ &\quad + \alpha(r_{j,t-1} - x_{1,j,t-1})(r_{j',t-1} - x_{1,j',t-1}).\end{aligned} \tag{10}$$

Here, $r_{j,t}$ is a rate of return of the j-th security realized at time t. We also notice that the above EMA definition indicates input variables $(x_{1,j,t}, x_{2,j,t}, x_{3,j,t})_{j=1,\ldots,N}$ at time t can be determined by the securities' rates of returns until time $t-1$.

As for the parameters of the membership functions $a_{i,t}$ and $b_{i,t}$, we adjust as follows:

$$a_{i,t} = \max_{j=1,\cdots,N}\{x_{i,j,t}\}, \quad b_{i,t} = \min_{j=1,\cdots,N}\{x_{i,j,t}\}. \tag{11}$$

Although in the current work, we decide these values in such an endogenous way, it is also possible to incorporate them into state variables, which is one of the most important future researches. Let us remark that the literature of evolving fuzzy systems [16]–[28] has provided systematic on-line algorithms for the membership functions' parameters of TSK models.

Now, we can practice the procedure of a zero-order TSK fuzzy system, explained in Section III-A, after deciding the parameters $(c_k)_k$ and obtaining time-series return data $(r_{j,t})_{j,t}$. Here, notice that our fuzzy system is expressed as a non-linear function $f(\cdot; Z_t) : \mathbb{R}^{3\times N} \to \mathbb{R}^N$, given a parameter vector Z_t. By applying this function $f(\cdot; Z_t)$ to the input variables $(x_{i,j,t})_{i,j} \in \mathbb{R}^{3\times N}$, the rating vector of the trading securities, denoted by $(y_{j,t})_j \in \mathbb{R}^N$, are obtained as the outputs of our fuzzy system.

However, for the practical implementation of asset allocation, there is an additional step to transform these ratings $(y_{j,t})_{j=1,\cdots,N}$ into a portfolio weight vector over the period $(t-1, t]$, denoted by $(\omega_{j,t})_{j=1,\cdots,N}$. Here, a portfolio weight of the j-th security $\omega_{j,t}$ is defined as a proportion of the amount of money invested in the j-th security to the total asset value (i.e. portfolio value) at $t-1$.

For transforming the rating values to portfolio weights $(\omega_{j,t})_{j=1,\cdots,N}$, we employ the following non-linear function $g : \mathbb{R}^N \to \mathbb{R}^N$:

$$\begin{aligned}(\omega_{j,t})_j &= g((y_{j,t})_j)\\ &= 1_{\{\sum_j y_{j,t} > 1\}}\frac{(y_{j,t})_j}{\sum_j y_{j,t}} + 1_{\{\sum_j y_{j,t} \le 1\}}(y_{j,t})_j,\end{aligned} \tag{12}$$

where

$$(y_{j,t})_j = f((x_{i,j,t})_{i,j}; (c_{k,t})_k). \tag{13}$$

Namely, the function g represents a portfolio weight constraint that the sum of the weights equals or less than one, which was referred in the beginning of this section as no leverage condition. As well, the restriction $c_k \in [0,1]$ $(k = 1, \cdots, 8)$ confirm no-short sale constraints. Thus, a resulting portfolio weight vector $(\omega_{j,t})_j$ satisfies

$$0 \leq \omega_{j,t}, \quad \sum_{j=1}^{N} \omega_{j,t} \leq 1, \tag{14}$$

where the remaining proportion, i.e. $1 - \sum_{j=1}^{N} \omega_{j,t}$ is invested in cash or bank account whose interest rate is supposed to be zero through time for conservative performance evaluation.

The next section shows a state space representation for learning our fuzzy system, which sequentially estimates the parameters $(c_k)_k$.

C. State Space Representation of Fuzzy System Learning

This section illustrates an example to learn the parameters $(c_k)_{k=1,\cdots,8}$ in the above fuzzy system, which are the most important variables for higher performance. In particular, we propose a state space representation of the parameter learning, which is designed to enhance the existing state-of-art technique, such as ANN, GA or PSO for fuzzy system identification.

Now, based on the notation of our state space framework described in Section II, the parameters $(c_k)_{k=1,\cdots,8}$ are interpreted as a state vector $Z_t = (c_{k,t})_{k=1,\cdots,8}$. In addition, the fuzzy system corresponds with the AI model $O_t = f(I_t; Z_t)$, where the input variables $I_t \in \mathbb{R}^{3\times N}$ are expected returns, volatilities and correlations for all the trading securities, while the output variable $O_t \in \mathbb{R}^N$ is a rating vector of the securities. Please remind that our fuzzy system is a non-linear mapping $f(\cdot; Z_t) : \mathbb{R}^{3\times N} \to \mathbb{R}^N$. Besides, there is also non-linearity for the parameters $(c_k)_k$ in our model owing to the portfolio constrains.

Specifically, we prepare the following state-space model, which will be estimated by the PF method.

$$\begin{aligned} \bar{r}_t &= \sum_{j=1}^{N} \omega_{j,t} r_{j,t} + u_t, && (15)\\ (\omega_{j,t})_j &= g \circ f((x_{i,j,t})_{i,j}; (c_{k,t})_k), && (16)\\ c_{k,t} &= h(\bar{c}_k + \phi(c_{k,t-1} - \bar{c}_k) + v_{k,t}), && (17) \end{aligned}$$

where $u_t \sim N(0, \sigma_u^2)$, $v_{k,t} \sim N(0, \sigma_k^2)$ and $\phi, (\bar{c}_k)_k$ are constant parameters $(k = 1, \cdots, 8)$, and $g \circ f$ denotes a composite function of $g : \mathbb{R}^N \to \mathbb{R}^N$ and $f(\cdot; Z_t) : \mathbb{R}^{3\times N} \to \mathbb{R}^N$. Please remember that g is a non-linear function for normalizing portfolio weights, as defined by Eq. (12) in Section III-B.

As described in Appendix, at each time step, the PF algorithm first propagates particles $Z_t^{[\ell]} (= (c_{k,t}^{[\ell]})_k)$, $\ell = 1, \cdots, L$ based on the state equation (17), and then resample them with weights proportional to their likelihoods $p(Y_t | Z_t^{[\ell]})$, $\ell = 1, \cdots, L$ calculated by the observation equation (15).

Therefore, in the following, let us explain the above state space model in reverse order, i.e. Eq. (17)- Eq. (15).

1) State Equation (17): First, the term $\bar{c}_k + \phi(c_{k,t-1} - \bar{c}_k) + v_{k,t}$ in the right-hand-side corresponds with a so-called autoregressive model of order one (AR(1) model). As is well known, the AR(1) model, also called a mean reversion process, tends to drift toward its long-term mean $\bar{c}_k$.

The current work proposes to decide the mean reversion levels $\bar{c}_k$ by implementing a particle swarm optimization (PSO) method over the training period. Namely, by using training data, PSO is firstly applied to learning the parameters $(c_k)_k$ in the fuzzy model f introduced in Section III-B. Then, over a test period, the parameters $(\bar{c}_k)_k$ in the state equation (17) is set to be the estimates of the parameters $(c_k)_k$ obtained by the PSO, whose detailed procedure will be explained in Section III-D3.

Let us notice that this modeling is expected to realize better estimation around the PSO results. Importantly, for all the reliable state-of-art methods including ANN, GA, PSO and so on, this approach is applicable to enhance model performance.

We also note the function $h : \mathbb{R} \to [0,1]$ represents the restriction that $(c_{k,t})_k$ take the values between 0 and 1, as follows.

$$h(c) = 1_{\{c>1\}} + 1_{\{0<c\leq 1\}} c, \tag{18}$$

where $1_{\{c>1\}}$ and $1_{\{0<c\leq 1\}}$ are indicator functions. Please recall this restriction was introduced in Section III-B for normalization.

In the PF algorithm at time step t, this state equation (17) generates $Z_t^{[\ell]} = (c_{k,t}^{[\ell]})_k$, $\ell = 1, \cdots, L$, which will be the candidates of the parameters $(c_{k,t})_k$ in our fuzzy system.

2) AI Model (16): In practical implementation of PF, by using the generated candidates of the parameters $(c_{k,t}^{[\ell]})_k$, $\ell = 1, \cdots, L$, our fuzzy system $f(\cdot; (c_{k,t}^{[\ell]})_k) : \mathbb{R}^{N\times 3} \to \mathbb{R}^N$ is applied to the inputs $(x_{i,j,t})_{i,j} \in \mathbb{R}^{3\times N}$ for each particle $\ell = 1, \cdots, L$.

As a result, we obtain the rating values of the securities $(y_{j,t}^{[\ell]})_{j=1,\cdots,N} \in \mathbb{R}^N$ for each $\ell = 1, \cdots, L$. Then, this rating vector $(y_{j,t}^{[\ell]})_{j=1,\cdots,N}$ is normalized by the function $g : \mathbb{R}^N \to \mathbb{R}^N$ to decide a portfolio weight vector $(\omega_{j,t}^{[\ell]})_{j=1\cdots,N}$ for a particle ℓ.

3) Observation Equation (15): As for the observation equation (15), we first note that a portfolio rate of return at time t is represented as the 1st term in the right-hand-side, $\sum_{j=1}^{N} \omega_{j,t} r_{j,t}$, which is derived as follows:

$$\sum_{j=1}^{N+1} \omega_{j,t}(1 + r_{j,t}) = 1 + \sum_{j=1}^{N+1} \omega_{j,t} r_{j,t} = 1 + \sum_{j=1}^{N} \omega_{j,t} r_{j,t}, \tag{19}$$

where $j = N+1$ denotes cash or bank account with a (lending) interest rate $r_{N+1,t} \equiv 0$ and a portfolio weight $\omega_{N+1,t} \equiv 1 - \sum_{j=1}^{N} \omega_{j,t}$ for all t, as mentioned in Section III-B.

That is, in the right-hand-side of the observation equation (15), after an investor constructs his/her portfolio according to the weight $(\omega_{j,t})_{j=1,\cdots,N}$ calculated from the fuzzy system at time $t-1$, its investment return realizes at time t as $\sum_{j=1}^{N} \omega_{j,t} r_{j,t}$. Remember that $r_{j,t}$ denotes a rate of return of j-th security realized at time t. Moreover, $\bar{r}_t$ in the left-hand-side stands for a target investment return at time t, which can be decided from each investor's preference.

In other words, the resampling step in the PF algorithm selects from the propagated particles $(\omega_{j,t}^{[\ell]})_j$, or $(c_{k,t}^{[\ell]})_k$, $\ell = 1, \cdots, L$ which realize the target portfolio return $\bar{r}_t$. Thus, the obtained filtered states $\hat{Z}_t^{[\ell']} = (\hat{c}_{k,t}^{[\ell']})_k$ $(\ell' = 1, \cdots, L)$ are utilized in the time-t implementation of the fuzzy system, which decides the portfolio weights $\omega_{j,t+1}$, as described in Section III-B. Precisely, we use the mean vector of $\hat{Z}_t^{[\ell']} = (\hat{c}_{k,t}^{[\ell']})_k$ over $\ell' = 1, \cdots, L$ as the estimates of $(c_k)_k$ at time t. Let us notice that we do not exploit the future information in the above procedure, that is, we only use the return data until time t to determine $(c_{k,t})_k$ and $(\omega_{j,t+1})_{j=1,\cdots,N}$.

Now, repeating this procedure sequentially updates the filtered states $\hat{Z}_t^{[\ell']} = (\hat{c}_{k,t}^{[\ell']})$ $(\ell' = 1, \cdots, L)$ through time by reflecting the past measurement (performance), which gives our fuzzy model high adaptability against the change of environment.

Also, let us mention to the possible extension of our asset allocation problems. Namely, although we set only target returns in the observation equation, it is not difficult to implement more complex objectives due to the general applicability of PFs. For instance, it seems interesting in practice to add a target risk term into Eq. (15), which also seems consistent with modern portfolio theory [35].

D. Numerical Experiment

This section shows a numerical experiment using actual time-series data of securities' returns.

1) Data: We use monthly total returns of the US and Japanese stock market indexes as well as the US treasury note as listed in Table II, where data source is Bloomberg. Hereafter we employ the abbreviations of the index names in this table. The time period of the return data is 329 months, from January 1990 to May 2017.

TABLE II
DATA

Index name	Ticker code (Bloomberg)	Abbreviation
S&P500 Index	SPTR.Index	S&P 500
Tokyo Stock Price Index	TPXDDVD.Index	TOPIX
S&P 10-Year U.S. Treasury Note Futures Total Return Index	SPUSTTR.Index	T-Note

A monthly security return $r_{j,t}$ is given by $r_{j,t} = (P_{j,t}/P_{j,t-1} - 1)$, where $P_{j,t}$ denotes the j-th security price at time t. Our data are downloaded in USD-denominated form so that we consider the global investment with no currency hedging, where the initial investment is made in USD.

2) Performance Measure: First of all, we define the portfolio values $\{V_t\}_{t=0,\cdots,T}$ and returns $\{R_t\}_{t=1,\cdots,T}$ as follows.

$$\begin{aligned} V_{t+1} &= V_t \left(1 + \sum_{j=1}^{N} \omega_{j,t+1} r_{j,t+1} \right) \\ &\quad - \sum_{j=1}^{N} d_j |\omega_{j,t+1} V_t - \omega_{j,t} V_{t-1}(1 + r_{j,t})|, \quad (20) \\ R_{t+1} &= V_{t+1}/V_t - 1, \quad (21) \end{aligned}$$

where d_j and $r_{j,t}$ denote a transaction spread and a return of j-th security, respectively, and the initial portfolio value is normalized to one, i.e. $V_0 = 1$

The penalty term $\sum_{j=1}^{N} d_j |\omega_{j,t+1} V_t - \omega_{j,t} V_{t-1}(1+r_{j,t})|$ of Eq. (20) is the total transaction cost arising from the allocation change at time t. Since $\omega_{j,t}$ and $\omega_{j,t+1}$ are portfolio weights of the j-th security during $[t-1, t)$ and $[t, t+1)$, $\omega_{j,t} V_{t-1}(1+r_{j,t})$ and $\omega_{j,t+1} V_t$ indicate the values of the j-th security before and after the position change at time t, respectively. That is, $|\omega_{j,t+1} V_t - \omega_{j,t} V_{t-1}(1+r_{j,t})|$ represents the necessary amount of money for the position change of the j-th security at time t. Hence, the total transaction cost at time t equals to the summation of $d_j |\omega_{j,t+1} V_t - \omega_{j,t-1} V_{t-1}(1 + r_{j,t})|$ for all j. In this paper, we set $d_j = 10\ bps$ $(j = 1, \cdots, N)$.

We evaluate the portfolio performances from various criteria; one return measure (compound return), three risk measures (standard deviation, downside deviation, maximum drawdown), and two risk-adjusted return measures (Sharpe ratio, Sortino ratio).

- Compound return (CR):

$$CR \equiv \left\{ \prod_{t=1}^{T} (1 + R_t) \right\}^{1/T} - 1. \quad (22)$$

  This is one of the most fundamental performance measures, which corresponds with a geometric average of the portfolio returns $\{R_t\}$ defined in Eq. (21).
- Standard deviation (SD):

$$SD \equiv \left\{ \frac{1}{T} \sum_{t=1}^{T} (R_t - \bar{R})^2 \right\}^{1/2}, \ \bar{R} \equiv \frac{1}{T} \sum_{t=1}^{T} R_t. \quad (23)$$

  This is one of the most basic variables both in theory and practice for portfolio risk management or derivatives pricing and hedging, which is also known as volatility.
- Downside deviation (DD):

$$DD \equiv \left\{ \frac{1}{T} \sum_{t=1}^{T} \min(0, R_t)^2 \right\}^{1/2} \quad (24)$$

  Differently from SD, this risk measure regards only negative return as risk, which seems reasonable for investment performance evaluation.
- Maximum drawdown (MDD):

$$MDD \equiv \max_{1 \le t \le T} \frac{M_t - V_t}{M_t}, \quad M_t \equiv \max_{0 \le s \le t} V_s. \quad (25)$$

  MDD is a famous concept in hedge fund risk management, where drawdown denotes a decline from the past peak value M_t to the present value V_t.
  Shortly, this measure tells us the worst senario for a given investment horizon. That is, it represents how much loss an investor suffers from if he/she enter and exit an investment at the worst timing.
  As it is widely recognized in practice that investment performance largely depends on its starting and exiting timing, MDD is thought to be an important measure. Namely, small MDD implies that an investor has not

suffered from a large loss, whenever he/she starts the investment, at least on the past data.

- Sharpe ratio (ShR):

$$ShR \equiv (\bar{R} - r_f)/SD, \tag{26}$$

where r_f denotes a risk-free rate. In investment performance evaluation, risk-adjusted returns are often regarded as the most important measures. Among them, ShR is the most famous one, which is also a basic quantity in the field of financial economics.
Here, we calculated a risk-free rate r_f from time series of one month dollar LIBOR (London interbank offered rate). Let us remark that r_f is interpreted as a borrowing interest rate on cash or bank account. As stated in Section III-B and III-C, its lending interest rate is assumed to be zero. We note that these asymmetric settings for interest rates are practical, as well as conservative in terms of performance evaluation.

- Sortino ratio (SoR):

$$SoR \equiv (\bar{R} - r_f)/DD. \tag{27}$$

SoR is also useful because it adjusts risk by using DD, which makes it possible to focus on only downside risk.

3) Numerical Result: In the subsequent numerical experiment, we specify the target returns introduced in Eq. (15) of Section III-C as follows.

$$\bar{r}_t = \max\{r_{1,t}, 0\}, \tag{28}$$

where the 1st index $i = 1$ denotes S&P 500, that is $r_{1,t}$ corresponds with the rate of return of S&P 500. This specification implies that the target of our fuzzy system is to realize S&P 500 index return with zero lower bound. Note that S&P 500 is one of the most classical and important stock market indexes especially for the US dollar-based investors.

We also remark that the payoff structure with zero lower bound is similar to the long position of a call option whose underlying security is S&P 500. Although the option has such a desirable payoff structure, its premium is too expensive to invest it in most cases. Therefore, this target return formulation is meaningful as is also tackled in [36], because our scheme is just a multi-securities' investment implementable without payment of any special premium.

As stated in Section III-C, we firstly implement PSO for obtaining the mean reversion levels $(\bar{c}_k)_{k=1,\cdots,8}$ from the about first half of data. Precisely, by using the period from January 1990 to December 2002 as training data, we apply PSO to the fuzzy system introduced in Section III-B for learning the parameters $(c_k)_{k=1,\cdots,8}$. Then, over the remaining period from January 2003 to May 2017, our PF method adaptively learns the state vector $Z_t = (c_{k,t})_k$ by setting the parameters $(\bar{c}_k)_{k=1,\cdots,8}$ to be the PSO estimates over the training data. Needless to say, investment performances will be evaluated only for the test period from January 2003 to May 2017.

TABLE III
RESULT OF PSO

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
0.214	0.057	0.731	0.765	0.000	0.000	0.860	0.566

The result of PSO for learning the parameters $(c_k)_{k=1,\cdots,8}$ is shown in Table III. Here, the fitness function of PSO is defined as

$$\text{Fitness} := -\frac{1}{T}\sum_{t=1}^{T}\left|\bar{r}_t - \sum_{j=1}^{N}\omega_{j,t}r_{j,t}\right|^2, \tag{29}$$

where the definitions of the variables and functions mentioned in Eq. (29) are introduced in Section III-C. As for the transition of each particle in the swarm, we adopt the basic type as follows: For $s = 1, 2, \cdots, S_{\max}$,

$$\begin{aligned} p_{s+1} &= p_s + v_{p,s+1}, \\ v_{p,s+1} &= \xi v_{p,s} + \epsilon_1(\hat{p}_s - p_s) + \epsilon_2(\hat{p}_{g,s} - p_s), \end{aligned} \tag{30}$$

where p_s and $v_{p,s}$ denote the location and velocity of particles in the swarm, while $\hat{p}_s$ and $\hat{p}_{g,s}$ represent the particle's best position and the swarm's best position, respectively. Further, $\epsilon_{s,1}$ and $\epsilon_{s,2}$ denote random numbers generated by uniform distribution $U(0, 1)$, respectively. We also set the constant parameter ξ to be 0.5. Moreover, the iteration is terminated if the swarm's best location $\hat{p}_{g,s}$ remains the same more than five times, where the number of particles in the swarm is 1,000,000. For the decaying parameter α in Eq. (10) and (11), we use so-called six months EMA, i.e. $\alpha = 2/(1 + 6)$.

Here, we briefly check the PSO results shown in Table III. First, we notice that c_5 and c_6 are estimated to be zeros. As stated in Section III-A, this result implies that 5th and 6th IF-THEN rules do not work in effect. Since the common part of these rules, i.e. (return is *low*) AND (volatility is *high*) shown in Table I, seems unnecessary to realize a target return defined by Eq. (28) under a no-short sale constraint with no leverage condition, the estimated results are quite natural. Next, we focus on the parameters c_3, c_4 and c_7 whose values are estimated to be high. Let us remember that our target return intends to replicate the return of S&P 500 with zero floor. Since low volatility and high correlation make it easier to replicate the target return, the high values of c_3 and c_7 seem to be reasonable. Also, the negative correlation and high expected return are effective for creating a zero floor, which leads to the high value of c_4. Overall, the estimate result is not against investment experts' intuition and knowledge.

TABLE IV
RESULT OF MEAN SQUARED ERROR: TRAINING PERIOD

PSO	B&H
0.058%	0.082%

Moreover, Table IV shows the following mean square errors (MSEs):

$$MSE = \frac{1}{T}\sum_{t=1}^{T} |\bar{r}_t - R_t|^2, \tag{31}$$

where $\bar{r}_t$ is target returns defined in Eq. (28), and R_t denotes investment returns defined in Eq. (21). Here, we set T to be the number of training periods. For comparison, we also show the MSE based on the S&P 500 buy-and-hold (B&H) strategy (i.e. $\omega_{1,t} \equiv 1$ for all t). Table IV tells that the learning with PSO outperforms the simple B&H strategy in terms of MSE.

Now, in the following, let us show the investment results over the test period. First of all, in running the PF, we set the exogenous parameters as follows: $\sigma_u = 0.01$, $\sigma_k = 0.1$, $\phi = 0.75$. Besides, the number of particles L are set to be 1,000,000.

Table V summarizes performance measures introduced in Section III-D2, whereas Table VI does MSEs. Here, for Table V, all the measures except maximum drawdown are presented in an annual basis.

In Table V, the column named by "PSO" presents the investment results based on the fuzzy system whose parameters $(c_k)_k$ are decided by the PSO estimates in Table III. Also, "Static 1" and "Static 2" denote traditional investment strategies, whose constant portfolio weights equal $\omega_t = (0.15, 0.15, 0.7)$ and $\omega_t = (0.35, 0.35, 0.3)$, respectively. Lastly, "S&P 500", "TOPIX" and "T-Note" represent buy-and-hold strategies for the trading securities, that is, the strategies with constant portfolio weights $\omega_t = (1, 0, 0), \omega_t = (0, 1, 0), \omega_t = (0, 0, 1)$.

It is observed from Table V that our proposed model improves the case of "PSO" for all the performance measures. Especially, risk-adjusted returns, i.e. Sharpe and Sortino ratios, are the highest in all the strategies. Note that fine risk-return efficiency, that is high risk-adjusted returns, is thought to be one of the most important features of sophisticated investment portfolios in financial industries.

Moreover, Table VI shows that our methodology successfully replicates the target returns, compared with PSO and B&H, that is, our model's MES is the lowest. Also, let us remark that differently from the training period, the MSE of PSO turns out to be worse than the simple B&H strategy in the test period, which implies the importance of tuning model parameters through time in the machine learning under the dynamic environment.

Overall, those results of the numerical experiment suggest that our Bayesian on-line learning approach is effective to improve performance of the existing PSO method, as we have intended in our state space modeling in Section III-C.

IV. Further Discussion

Although in the current numerical experiment, the proposed state space model has been applied to parameter learning of a fuzzy system, it is also applicable to the dynamic model selection as mentioned in Section II.

Particularly, in this example, we have considered three input variables (expected return, volatility and correlation) with two fuzzy sets (*high* and *low*), which generates eight IF-THEN rules as in Table I. However, if the number of input variables and/or fuzzy sets increases, the number of corresponding rules gets exponentially large, which will make parameter estimation unreliable.

In these cases, it seems effective to utilize a dynamic model selection approach. That is, we firstly prepare several fuzzy systems, each of which has a small number of promising rules. Here, we suppose that the parameters such as (c_k) are decided by the expert knowledge, whose assumption is not necessarily indispensable as explained later. Then, these multiple patterns of fuzzy systems are dynamically selected with our filtering methodology.

Concretely, for a given M number of candidate fuzzy systems, a dynamic model selection method can be implemented by introducing an integer-valued state variable $Z_t \in \{1, 2, \cdots, M\}$, which represents indexes of the candidate models. Then, one of the effective formulations of state transition for the model index Z_t is a Markov switching process.

$$P(Z_t = m | Z_{t-1} = n) = p_{n,m}, \quad m, n \in \{1, 2, \cdots, M\}, \tag{32}$$

where $p_{n,m}$ denotes a transition probability from model n to model m. For the example of financial investment, as a market phase such as uptrend, downtrend or range phases drastically changes through time, it is promising to prepare fuzzy systems performing well in different market phases and apply the above method.

Here, we remark that it is not inevitable to strictly decide the parameter values of the fuzzy systems in advance. With a state vector representing the model parameters as well as model indexes, dynamic parameter estimation and model selection can be carried out at the same time.

Further, it is also quite important that the applicability of our methodology is not limited to fuzzy systems. That is, we can use other AI models including linear regression/classification models, support vector machines or ANNs. For example, although network structure of ANNs such as the number of layers or nodes strongly affects their performances, the optimal numbers of layers and nodes are not known in general. Then, by preparing various kinds of ANNs with different numbers of layers and nodes, our proposed method is expected to adaptively choose a proper model from those candidates.

Lastly, we notice that this approach possibly shows a new direction in the realm of Bayesian model selection and comparison for machine learning, that is, it is a different approach from the typical model evidence-based ones. (Please see a textbook [37] for the standard theory of Bayesian model selection.)

V. Conclusion

In this paper, we have proposed a new state space approach to adaptive fuzzy modeling under the dynamic environment, in which the model parameters as well as model structures are represented by time-varying state variables. Especially, our approach intends to stably improve outcomes based on

TABLE V
RESULT OF INVESTMENT

	Our model	PSO	Static 1	Static 2	S&P 500	TOPIX	T-Note
Compound return	0.082	0.072	0.061	0.075	0.094	0.067	0.048
Standard deviation	0.057	0.058	0.052	0.089	0.136	0.153	0.060
Downside deviation	0.030	0.033	0.032	0.059	0.091	0.098	0.034
Maximum drawdown	0.061	0.103	0.094	0.320	0.510	0.469	0.065
Sharpe ratio	1.155	0.975	0.881	0.695	0.624	0.404	0.562
Sortino ratio	2.169	1.721	1.447	1.044	0.934	0.627	0.982

TABLE VI
RESULT OF MEAN SQUARED ERROR: TEST PERIOD

Our model	PSO	B&H
0.054%	0.112%	0.069%

the existing learning algorithms by employing mean-reversion processes for state equations.

Our approach consists of two procedures: First, we calculate the mean-reversion levels by the existing method, i.e. PSO, from training data. Then, in the test period, the PF implementation enables to obtain on-line estimates adaptive to the environmental changes.

To show the validity of our scheme, we have constructed a fuzzy system for asset allocation, because financial markets are typical examples of the dynamic environment. As a result, an out-of-sample numerical experiment has confirmed that our approach successfully enhances the PSO results in terms of all the evaluation criteria, which also attains the higher investment performance compared to traditional static strategies.

APPENDIX
ALGORITHM OF PARTICLE FILTER

In this paper, we utilize the following classical PF algorithm developed by [38], [39].

First of all, Bayes formula shows

$$p(Z_t|Y_{1:t}) \propto p(Y_t|Z_t)p(Z_t|Y_{1:t-1}). \tag{33}$$

Then, we apply a sampling importance resampling (SIR) method to obtain the samples $\{\hat{Z}_t^{[\ell]}\}_{\ell=1,\cdots,L}$ from the posterior $p(Z_t|Y_{1:t})$. That is, by regarding the prior $p(Z_t|Y_{1:t-1})$ as an importance function, we first draw $\{Z_t^{[\ell]}\}_{\ell=1,\cdots,L}$ from the distribution $p(Z_t|Y_{1:t-1})$, then resample from them with weights proportional to the likelihood $p(Y_t|Z_t)$, which gives us $\{\hat{Z}_t^{[\ell]}\}_{\ell=1,\cdots,L}$. These random samples are called "particles" in the PF literature.

Here, we get the samples $\{Z_t^{[\ell]}\}_{\ell=1,\cdots,L}$ from the prior $p(Z_t|Y_{1:t-1})$ by the following way. Note that

$$\begin{aligned} p(Z_t|Y_{1:t-1}) &= \int\int p(Z_t, Z_{t-1}, v_t|Y_{1:t-1})dZ_{t-1}dv_t \\ &= \int\int p(Z_t|Z_{t-1}, v_t)p(v_t)p(Z_{t-1}|Y_{1:t-1})dZ_{t-1}dv_t \\ &= \int\int \delta(Z_t - F(Z_{t-1}, v_t))p(v_t)p(Z_{t-1}|Y_{1:t-1})dZ_{t-1}dv_t, \end{aligned} \tag{34}$$

where $\delta(\cdot)$ is the Dirac delta function. Then, for given the samples of $\{v_t^{[\ell]}\}_{\ell=1,\cdots,L}$ and $\{\hat{Z}_{t-1}^{[\ell]}\}_{\ell=1,\cdots,L}$ from $p(v_t)$ and $p(Z_{t-1}|Y_{1:t-1})$, we obtain the samples $\{Z_t^{[\ell]}\}_{\ell=1,\cdots,L}$ from $p(Z_t|Y_{1:t-1})$ by setting $Z_t^{[\ell]} = F(\hat{Z}_{t-1}^{[\ell]}, v_t^{[\ell]})$.

Thus, the PF algorithm is summarized as follows:

(i) Generate the initial state vector $\{\hat{Z}_0^{[1]}, \cdots, \hat{Z}_0^{[L]}\}$.

(ii) Apply the following steps (ii-a)∼(ii-d) to each time $t = 1, \cdots, T$.

(ii-a) Generate system noise $v_t^{[\ell]}$, $\ell = 1, \cdots, L$.

(ii-b) Compute for each $\ell = 1, \cdots, L$ $Z_t^{[\ell]} = F(\hat{Z}_{t-1}^{[\ell]}, v_t^{[\ell]})$.

(ii-c) Evaluate the weights of particles $\{Z_t^{[1]}, \cdots, Z_t^{[L]}\}$ as $\delta_t^{[\ell]} \equiv p(Y_t|Z_t^{[\ell]})$, $\ell = 1, \cdots, L$ by using the likelihood function.

(ii-d) Resample $\{\hat{Z}_t^{[1]}, \cdots, \hat{Z}_t^{[L]}\}$ from $\{Z_t^{[1]}, \cdots, Z_t^{[L]}\}$. More precisely, resample each $\hat{Z}_t^{[\ell']}$, $\ell' = 1, \cdots, L$ from $\{Z_t^{[1]}, \cdots, Z_t^{[L]}\}$ with the probability given by

$$\text{Prob.}(\hat{Z}_t^{[\ell']} = Z_t^{[\ell]}|Y_t) = \frac{\delta_t^{[\ell]}}{\sum_{m=1}^{L}\delta_t^{[m]}}, \ \ell = 1, \cdots, L.$$

Here, the likelihood at time t, $p(Y_t|Z_t)$, is approximately calculated by

$$p(Y_t|Z_t) \simeq \frac{1}{L}\sum_{\ell=1}^{L}\delta_t^{[\ell]}. \tag{35}$$

REFERENCES

[1] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, p. 452, 2015.

[2] D. E. Rumelhart, G. E. Hinton, R. J. Williams *et al.*, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

[3] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[4] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[6] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*. Springer, 2011, pp. 760–766.

[7] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[8] S.-I. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," *IEEE transactions on Neural Networks*, vol. 3, no. 5, pp. 801–806, 1992.

[9] J.-S. Jang, "Anfis: adaptive-network-based fuzzy inference system," *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[10] M. A. Lee and H. Takagi, "Integrating design stage of fuzzy systems using genetic algorithms," in *Fuzzy Systems, 1993., Second IEEE International Conference on*. IEEE, 1993, pp. 612–617.

[11] C. L. Karr and E. J. Gentry, "Fuzzy control of ph using genetic algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 1, p. 46, 1993.

[12] F. Herrera and J. L. Verdegay, *Genetic algorithms and soft computing*. Physica-Verlag, 1996.

[13] H. Zhenya, W. Chengjian, Y. Luxi, G. Xiqi, Y. Susu, R. C. Eberhart, and Y. Shi, "Extracting rules from fuzzy neural network by particle swarm optimisation," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. IEEE, 1998, pp. 74–77.

[14] I. Pan and D. Bester, "Fuzzy bayesian learning," *IEEE Transactions on Fuzzy Systems*, 2017.

[15] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International journal of man-machine studies*, vol. 7, no. 1, pp. 1–13, 1975.

[16] P. Angelov and R. Buswell, "Evolving rule-based models: A tool for intelligent adaptation," in *IFSA world congress and 20th NAFIPS international conference, 2001. Joint 9th*, vol. 2. IEEE, 2001, pp. 1062–1067.

[17] P. Angelov, C. Xydeas, and D. Filev, "On-line identification of mimo evolving takagi-sugeno fuzzy models," in *Fuzzy Systems, 2004. Proceedings. 2004 IEEE International Conference on*, vol. 1. IEEE, 2004, pp. 55–60.

[18] P. Angelov and D. Filev, "Simpl_ets: A simplified method for learning evolving takagi-sugeno fuzzy models," in *Fuzzy Systems, 2005. FUZZ'05. The 14th IEEE International Conference on*. IEEE, 2005, pp. 1068–1073.

[19] P. P. Angelov, *Evolving rule-based models: a tool for design of flexible adaptive systems*. Physica, 2013, vol. 92.

[20] E. D. Lughofer, "Flexfis: A robust incremental learning approach for evolving takagi–sugeno fuzzy models," *IEEE Transactions on fuzzy systems*, vol. 16, no. 6, pp. 1393–1410, 2008.

[21] N. Kasabov, "Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 31, no. 6, pp. 902–918, 2001.

[22] N. K. Kasabov and Q. Song, "Denfis: dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 144–154, 2002.

[23] S. Wu, M. J. Er, and Y. Gao, "A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 578–594, 2001.

[24] G. Leng, T. M. McGinnity, and G. Prasad, "An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network," *Fuzzy sets and systems*, vol. 150, no. 2, pp. 211–243, 2005.

[25] E. Lima, F. Gomide, and R. Ballini, "Participatory evolving fuzzy modeling," in *Evolving Fuzzy Systems, 2006 International Symposium on*. IEEE, 2006, pp. 36–41.

[26] L. Maciel, A. Lemos, F. Gomide, and R. Ballini, "Evolving fuzzy systems for pricing fixed income options," *Evolving Systems*, vol. 3, no. 1, pp. 5–18, 2012.

[27] L. Maciel, F. Gomide, and R. Ballini, "Enhanced evolving participatory learning fuzzy modeling: an application for asset returns volatility forecasting," *Evolving Systems*, vol. 5, no. 2, pp. 75–88, 2014.

[28] L. Maciel, R. Vieira, A. Porto, F. Gomide, and R. Ballini, "Evolving participatory learning fuzzy modeling for financial interval time series forecasting," in *Evolving and Adaptive Intelligent Systems (EAIS), 2017*. IEEE, 2017, pp. 1–8.

[29] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE transactions on systems, man, and cybernetics*, no. 1, pp. 116–132, 1985.

[30] M. Sugeno and G. Kang, "Structure identification of fuzzy model," *Fuzzy sets and systems*, vol. 28, no. 1, pp. 15–33, 1988.

[31] M. Nakano, A. Takahashi, and S. Takahashi, "Generalized exponential moving average (ema) model with particle filtering and anomaly detection," *Expert Systems with Applications*, vol. 73, pp. 187–200, 2017.

[32] A. M. Rather, A. Agarwal, and V. Sastry, "Recurrent neural network and a hybrid model for prediction of stock returns," *Expert Systems with Applications*, vol. 42, no. 6, pp. 3234–3241, 2015.

[33] C.-J. Huang, D.-X. Yang, and Y.-T. Chuang, "Application of wrapper approach and composite classifier to the stock trend prediction," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2870–2878, 2008.

[34] M. Nakano, A. Takahashi, and S. Takahashi, "Fuzzy logic-based portfolio selection with particle filtering and anomaly detection," *Knowledge-Based Systems*, 2017.

[35] H. Markowitz, "Portfolio selection," *The journal of finance*, vol. 7, no. 1, pp. 77–91, 1952.

[36] M. Nakano, A. Takahashi, and S. Takahashi, "Creating investment scheme with state space modeling," *Expert Systems with Applications*, vol. 81, pp. 53–66, 2017.

[37] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[38] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEE Proceedings F (Radar and Signal Processing)*, vol. 140, no. 2. IET, 1993, pp. 107–113.

[39] G. Kitagawa, "Monte carlo filter and smoother for non-gaussian nonlinear state space models," *Journal of computational and graphical statistics*, vol. 5, no. 1, pp. 1–25, 1996.