Variational learning for Generalized Associative Functional Networks in modeling dynamic process of plant growth

Han-Bing Qu a,⁎, Bao-Gang Hu b,c

a Beijing Research Center For Pattern Recognition Technology, Beijing Academy of Science and Technology, Beijing 100012, China
b National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, P.O. Box 2728, Beijing, 100080, China
c Beijing Graduate School, Chinese Academy of Sciences, Beijing 100049, China

A R T I C L E   I N F O

Article history:
Received 16 December 2007
Received in revised form 17 June 2009
Accepted 18 June 2009

Keywords:
Functional equations
Generalized Associative Functional Networks
Variational Bayes
Bayesian backfitting
Dynamic plant growth modeling

A B S T R A C T

This paper presents a new statistical techniques — Bayesian Generalized Associative Functional Networks (GAFN), to model the dynamical plant growth process of greenhouse crops. GAFNs are able to incorporate the domain knowledge and data to model complex ecosystem. By use of the functional networks and Bayesian framework, the prior knowledge can be naturally embedded into the model, and the functional relationship between inputs and outputs can be learned during the training process. Our main interest is focused on the Generalized Associative Functional Networks (GAFNs), which are appropriate to model multiple variable processes. Three main advantages are obtained through the applications of Bayesian GAFN methods to modeling dynamic process of plant growth. Firstly, this approach provides a powerful tool for revealing some useful relationships between the greenhouse environmental factors and the plant growth parameters. Secondly, Bayesian GAFN can model Multiple-Input Multiple-Output (MIMO) systems from the given data, and presents a good generalization capability from the final single model for successfully fitting all 12 data sets over 5-year field experiments. Thirdly, the Bayesian GAFN method can also play as an optimization tool to estimate the interested parameter in the agro-ecosystem. In this work, two algorithms are proposed for the statistical inference of parameters in GAFNs. Both of them are based on the variational inference, also called variational Bayes (VB) techniques, which may provide probabilistic interpretations for the built models. VB-based learning methods are able to yield estimations of the full posterior probability of model parameters. Synthetic and real-world examples are implemented to confirm the validity of the proposed methods.

1. Introduction

A number of statistical modeling techniques have been widely used for ecological modeling in recent years, for example, artificial neural networks (ANN), fuzzy method, generalized linear model (GLM) and generalized additive model (GAM) (Džeroski and Drummrb, 2003; Guisan and Zimmermann, 2000; Okeke and Karnieli, 2006; Rechnagel, 2001; Yao et al., 2006; Zhang et al., 2005). These methodologies have some desirable characteristics: they are robust while applied to correlated data, have less restrictions in assumptions, are flexible in modeling non-linearity and non-constant variance structures, and have strong predictive abilities (Guisan et al., 2002; Olden and Jackson, 2002; Park et al., 2003; Zhang et al., 2005). Among all these techniques, neural network methods are especially receiving greater attention in the ecological sciences as modeling tools (Lek and Guégan, 1999; Özesmi et al., 2006; Paruelo and Tomasel, 1997; Schultz et al., 2000). Furthermore, the interpretation of the trained neural networks gains more concerns to ecologists, since the explanatory power of statistical models is desirable for obtaining knowledge of the causal relationships between inputs and outputs (Gevrey et al., 2003; Olden and Jackson, 2002; Olden et al., 2004). More statistical methods are developed to extract or embed knowledge for the data-driven models (Džeroski and Todorovski, 2003; Hu et al., 2005, 2007; Thompson and Kramer, 1994). In this paper, we propose another typical neural-based paradigm: Functional Networks, which can be regarded as a generalization of conventional neural networks, for the statistical ecological modeling. In this work, we make effort to embed prior knowledge from the start in contrast with the neural networks. That is, by use of the functional equation, we simplify the network structure to functional networks through functional constraints. In a way, the prior knowledge is embedded into the process of modeling.

☆ This work is supported in part by the Natural Science Foundation of China (#60275025, #60121302) and Chinese 863 Program (#2002AA241221).

† Special thanks to E. Heuvelink for his generous help to provide the experimental data.

⁎ Corresponding author. Tel.: +86 10 84935056.
E-mail addresses: quhanbing@gmail.com (H.-B. Qu), hubg@nlpr.ia.ac.cn (B.-G. Hu).

© 2009 Elsevier B.V. All rights reserved.
When the functional networks are used for modeling, an algorithm should be provided for the learning of model parameters. As a state-of-art approach, variational Bayesian approach is our choice to estimate the distribution of parameters.

Functional equations are basic mathematical equations along with differential equations, integral equations and other similar equations. The theory of functional equations, which could date back to 1747 from the papers of D’Alembert (Aczél, 1966), has received an increasing attention in both scientific studies as well as engineering applications. A recent development of the functional equations is a neural-based paradigm — Functional Networks (FN). Most of the neural networks are treated as general approximation tools and do not allow for the functional structure and properties of the function being modeled (Castillo et al., 1999). On the contrary, FN models not only bring the domain knowledge into consideration to determine the structure of networks, but also take data into account in estimating the unknown neuron functions for the modeling. One important advantage of FN model is the possibility to deal with functional constraints which are determined by functional properties which may be known previously, for example, associativity and some input–output relationships for some arguments. Another advantage is that FN allows arbitrary neural functions as long as the neural functions fulfill the functional solution conditions. All in all, FN models provide an easy approach for engineers and scientists to make use of the theories of functional equations which are hard to be understood (Castillo and Gutiérrez, 1998; Castillo et al., 2001; El-Sebakhy, 2004; Fontenla-Romero et al., 2004; Lacruz et al., 2006).

Bayesian data analysis is a practical method for making inferences from data using probabilistic models for quantities one observes and for quantities about which one wishes to learn. Gelman et al., 1995; Girolami and Rogers, 2006). Unlike the maximum likelihood and maximum a posteriori (MAP) approaches, which provide a single point estimation in parameter space, Bayesian inference aims to model entirely the posterior probability distribution (Gregory, 2005; Nasios and Bors, 2006). Bayesian methods permit several advantages: Bayesian approaches do not suffer from overfitting and have good generalization capabilities, prior knowledge can be easily incorporated and uncertainty is manipulated in a consistent way, and they achieve models of lower complexity (Attias, 2000; Beal and Ghahramani, 2003; Bishop, 2006; Jaakkola and Jordan, 2000; Ji et al., 2006). Main strategies in Bayesian inference consist of setting up a full probability model, estimating the posterior distributions of parameters and integrating over the entire parameter space. However, the integration in high-dimensional parameter space may be intractable and computationally expensive. Three approaches are often used to estimate the full posterior distributions of parameter and integrate over the entire parameter space: Laplacian method, Markov Chain Monte Carlo (MCMC), and variational learning (Gelman et al., 1995; Ghahramani and Beal, 2000; Husmeier, 2000; Nasios and Bors, 2006). The Laplacian method approximates the expression of the integrals by use of Taylor expansion. The disadvantage of this approach is that the approximation may become deteriorated and is computational intensive in high dimensions. MCMC is a kind of sampling framework for the estimation and integration of posterior distributions of parameters. Although MCMC methods may provide true posterior distributions of parameters, they are computationally time consuming and assessment of convergence is often problematic (Husmeier, 2000; Roberts and Penny, 2002). Variational Bayesian algorithm optimizes a lower bound to approximate the marginal likelihood in a procedure similar to the standard Expectation–Maximization (EM) algorithm (Beal and Ghahramani, 2006). Unlike the MCMC sampling strategy, variational learning algorithm is a deterministic and parametric approximation procedure for the posterior distributions over parameters (Bishop, 2006; Jaakkola and Jordan, 2000). Variational learning has been applied to many statistical models, including the factor analysis, independent component analysis and generalized autoregressive models (Jaakkola and Jordan, 2000; Ghahramani and Beal, 2000; Choudrey et al., 2000; Nasios and Bors, 2006; Honkela and Valpola, 2004; Roberts and Penny, 2002). In this paper, we propose variational algorithms for functional networks, and illustrate how they can be used in agro-ecological modeling.

The remainder of this paper is organized as follows. In the Section 2, we describe the basic theories of functional equations and associative functional network. Thereafter, we explain the fundamental idea of the variational approximation procedure and describes how the Generalized Associative Functional Networks (GAFN) can be treated as probabilistic graphical models. Under the variational Bayesian framework, we also provides two variational algorithms for the learning of GAFN model. Section 3 describes detail information about the synthetic and real data sets used in this paper. Section 4 illustrates how Bayesian GAFN is applied to the estimation and approximation problems on the synthetic and real-world models. Finally, we summary and discuss some further directions.

2. Method

2.1. Generalized Associative Functional Networks

Functional equations have a long history which could date back to the three papers of J. D’Alembert in 18th century, and play an important role in the development of mathematical analysis and probability theory (Aczél, 1966; Jaynes, 2003). A commonly-used functional equation is the generalized associative functional equations, which can be expressed as (Castillo and Ruiz-Cobo, 1992)

\[ F[G(x_1, x_2), x_3] = K[x_1, N(x_2, x_3)]. \]

The functional networks can be illustrated in Fig. 1. One may refer to Aczél (1966), Castillo and Ruiz-Cobo (1992) and Castillo et al. (1999) for more detailed description about functional equations and functional networks.

If the model has a d-dimensional input vector x and a one dimensional output variable which is denoted by y, GAFN model can be written as

\[ y = f_d + 1[f_1(x_1) + f_2(x_2) + ... + f_d(x_d)] \]

where \( f_{d + 1} \) is a strictly monotonic functional. This means \( f_{d + 1} \) is an invertible function. Without loss of generalization, we restrict the

![Fig. 1. The solution process of associative functional equation. (a) Initial network topology of associative functional equation, and (b) the network structure of the solution for associative functional equation.](Image)
problem to be Multi-Input Single-Output (MISO) model, $f_i(x_i)(i = 1, ..., d + 1)$ are all single input variables that fulfill the solution conditions. Usually, the true forms of $f_i(x_i)$ may not be known in advance, thus these functions can be approximated by

$$f_i(x_i) = \sum_{m=1}^{M_i} w_{im} \phi_{im}(x_i), \quad i = 1, ..., d + 1$$ (3)

where $\phi_{im}(x_i)(m = 1, ..., M_i)$ are linearly independent approximation functions for the $i$th variable, and $w_{im}(m = 1, ..., M_i)$ are weights of approximate functions. $M_i(i = 1, ..., d + 1)$ are the orders of basis functions for the $i$th variable. Fig. 2 illustrates the approximation structure of GAFN model. Because the $f_{d+1}$ is an invertible functional, thus the approximate networks for Eq. (2) can also be written as

$$f_{d+1}(y) = f_1(x_1) + f_2(x_2) + ... + f_d(x_d).$$ (4)

$f_{d+1}(y)$ can be assumed to be a functional of known form as long as it is a strictly monotonic function, such as the identity function, log function, or logit function.

2.2. Parallel algorithm for Bayesian GAFN

In the first place, we introduce a parallel algorithm for the variational learning of GAFN. If the observed data set is denoted by $D = \{(x_1, y_1), ..., (x_N, y_N)\}$, where each data $x$ is a $d$-dimensions vector, and the elements of $x$ have associative property, one can form GAFN model for these data according to Eq. (4)

$$\hat{f}_1(x_1) + \hat{f}_2(x_2) + ... + \hat{f}_d(x_d) + \hat{f}_{d+1}(x_{d+1}) = 0.$$ (5)

Here, $x_{d+1}$ denotes the output variable instead of $y$. The negative sign and inverse function symbol have been absorbed into each function for formal expression. If the approximate basis functions of Eq. (3) are used, thus the model can be written as

$$\Theta = \sum_{i=1}^{d+1} \sum_{m=1}^{M_i} w_{im} \phi_{im}(x_i),$$ (6)

where $\Theta$ is a hypothetical observed variable, $\phi_{im}(x_i)(m = 1, ..., M_i, i = 1, ..., d + 1)$ are the bases of approximation functions for each variable and $M_i (i = 1, ..., d + 1)$ are the basis orders of the approximation functions. Therefore, GAFN provides an approximation model for the obtained data. For the uniqueness of representation, one also has to specify the values of approximation functions at a point (one may refer to Castillo and Gutiérrez, 1998; Castillo et al., 1999) for detailed discussions), that is:

$$\hat{f}_i(x_i) = \sum_{m=1}^{M_i} w_{im} \phi_{im}(x_i) = \xi_i, \quad i = 1, ..., d + 1.$$ (7)

If additive noise $\varepsilon$ is assumed, which pertains Gaussian distribution with zero mean and with precision parameter (inverse variance) $\beta$, thus the matrix form for the GAFN model can be formulated as $(\Theta, \Phi(x)w, \beta^{-1})$ and its probabilistic model can be written as

$$p(\Theta, \Phi(x)w, \beta^{-1}) = N(\theta; \Phi(x)w, \beta^{-1}).$$ (8)

where $N(\theta; \Phi(x)w, \beta^{-1})$ is normal distribution with mean $\Phi(x)w$ and precision $\beta^{-1}$, and weight vector $w$ is subject to the equality constraint

$$\Phi(x_i)w = \zeta.$$ (9)

If one set $M = M_1 + ... + M_d + 1$, then $w$ is an $M \times 1$ vector, $\xi$ is a $(d + 1) \times 1$ vector and $\Phi(x_0)$ is a $(d + 1) \times M$ matrix. Following Rao (2002), an equivalent probabilistic model $(Z, \Phi(x)\Theta, \beta^{-1})$ can be built by solving Eq. (9), where $\Phi(x_i)\Theta = 0$ and $\Theta$ is an arbitrary random parameter vector consisting of $(M - d - 1)$ elements. This result can be explained as follows. Suppose $w_0 + \Theta \Theta$ is a general solution of $\Phi(x_0)w = \xi$, where $w_0$ is a particular solution of Eq. (9) and $B$ satisfies $\Phi(x_0)B = 0$. $\Theta$ is a new $(M - d - 1) \times 1$ parameter vector. Therefore, one has the following equivalent models (Qu and Hu, 2007)

$$E(\Theta) = \Phi(x)w = \Phi(x)(w_0 + \Theta \Theta)$$ or
$$E(\Theta) - E(\Theta|w_0) = \Theta \Theta.$$ (10)

Let $Z = \Theta - \Phi(x)w_0$, then $E(Z) = \Theta \Theta$. Therefore, the probabilistic model (8), which is subject to equality constraint (9), is reduced to a new equivalent probabilistic model

$$p(Z|x, \theta, \beta^{-1}) = N(Z; \Phi(x)\Theta, \beta^{-1}).$$ (11)

After the transformation, the equality constraint on GAFN model is vanished. In order to use the variational Bayes (VB) approximation procedure, the parameters $\Theta$ are assumed to be drawn from a zero-mean Gaussian prior with an isotropic covariance having precision $\alpha$. The weight precision $\alpha$ and noise precision $\beta$ are assumed to be drawn from Gamma priors with parameters $(\alpha_0, \alpha_0) \alpha_0$ and $(\beta_0, \beta_0)$ (MacKay, 1997; Roberts and Penny, 2002; Bishop, 2006).

$$p(\Theta|\alpha) = N(\Theta; 0, \alpha^{-1}).$$ (12)

$$p(\alpha) = \mathcal{G}(\alpha|\alpha_0, \alpha_0).$$ (13)

$$p(\beta) = \mathcal{G}(\beta|\beta_0, \beta_0).$$ (14)

The marginal likelihood or model evidence $p(D)$ of the probabilistic model is the likelihood of the model after its parameters have been integrated out from their joint distribution $p(D, \Theta, \alpha, \beta)$, which can be written as

$$\log p(D) = \log \int q(\Theta, \alpha, \beta) p(D, \Theta, \alpha, \beta) d\Theta d\alpha d\beta,$$ (15)

where $q(\Theta, \alpha, \beta)$ is the approximating posterior distribution. Eq. (15) can be decomposed into two items

$$\log p(D) = F + KL(q || p),$$ (16)
where \( F = \int q(\theta, \alpha, \beta) \log \frac{p(D, \theta, \alpha, \beta)}{q(\theta, \alpha, \beta)} d\theta d\alpha d\beta \),

\[
KL(q \mid \mid p) = - \int q(\theta, \alpha, \beta) \log \frac{p(\theta, \alpha, \beta \mid D)}{q(\theta, \alpha, \beta)} d\theta d\alpha d\beta.
\]

where the negative \( F \) is known as the variational free energy also called the Helmholtz energy, which is variational lower bound on the log marginal likelihood (Woolrich and Behrens, 2006; Beal et al., 2003; Ji et al., 2006). \( KL(q \mid \mid p) \) is the Kullback–Leibler (KL) divergence or relative entropy between the approximating posterior distribution \( q(\theta, \alpha, \beta) \) and true posterior distribution \( p(\theta, \alpha, \beta \mid D) \). Note that \( KL(q \mid \mid p) \geq 0 \), having an equality if, and only if, \( p(\theta, \alpha, \beta \mid D) = q(\theta, \alpha, \beta) \) (Beal and Ghahramani, 2006). The KL-divergence is strictly positive which means that \( F \) is a strict lower bound on the log evidence. By maximizing \( F \), we can minimize the KL-divergence between the approximating and true posterior. An appropriate form of the \( q(\theta, \alpha, \beta) \) leads to a tractable Bayesian learning.

Therefore, the approximation posterior distribution \( q(\theta, \alpha, \beta) \) over parameters \( \theta \) could be factorized into \( q(\theta, \alpha, \beta) = q(\theta)q(\alpha)q(\beta) \) (Ghahramani and Beal, 2000). The parallel algorithm for Bayesian GAFN is shown in Algorithm 1.

**Algorithm 1.** Parallel learning for Bayesian GAFN

**Transformation:**
- Solve \( \Phi(x_0)w = \xi \rightarrow w_0, \theta \) and \( \Phi(x_0)B = 0 \rightarrow B \);
- Transfer \( (\Theta, \Phi(x)\mathbf{w}, \beta^{-1}) \rightarrow (Z, \Phi(x)B\Theta, \beta^{-1}) \);

**Initialize:**
- \( \theta, \alpha_0, \beta_0, F_{\text{old}} \), tolerance;
- \( \Delta_{\text{old}} = \alpha_{\text{old}}^T \alpha_{\text{old}} = \beta_{\text{old}}^T \beta_{\text{old}}, F_{\text{new}} \);

while \( (\Delta F) > \) tolerance do

- Update the hyperpriors of \( \alpha \) and \( \beta \) using
  \[
  c_{\alpha}^N = \frac{(M - d - 1)}{2} + c_{\alpha}^0, \quad c_{\beta}^N = \frac{N}{2} + c_{\beta}^0
  \]
  \[
  d_{\alpha}^a = \frac{1}{2} (c_{\alpha}^0 + 1) \quad \text{and} \quad d_{\beta}^a = \frac{1}{2} (c_{\beta}^0 + 1)
  \]
  \[
  \hat{\alpha} = c_{\alpha}^N / d_{\alpha}^a, \quad \hat{\beta} = c_{\beta}^N / d_{\beta}^a
  \]

with

\[
E_{\hat{\Theta}} (\hat{\Theta}) = E_D (\hat{\Theta}) + \frac{1}{2} \text{Tr} \left( \hat{B}^T \Phi(X)^T \Phi(X) B \right)
\]

- Update the mean and variance of the parameters using
  \[
  \hat{\Sigma} = \left[ \hat{\beta} \hat{B}^T \Phi(X)^T \Phi(X) B + \alpha I \right]^{-1} \hat{\Theta} = \hat{B}^T \Phi(X)^T \beta Y
  \]
  \[
  \text{Calculate the negative free energy, } F_{\text{new}} \text{ using}
  \]
  \[
  F = L(q) - KL(\Theta) - KL(\alpha) - KL(\beta),
  \]
  \[
  L(q) = \frac{N}{2} \left( \Psi \left( \alpha_{\text{old}}^N \right) - \log d_{\alpha}^a \right) - \frac{1}{2} \text{Tr} (\hat{\Theta} - \hat{\Sigma})
  \]
  \[
  \text{Calculate the } \Delta F = |F_{\text{new}} - F_{\text{old}}| ;
  \]

end while

Post - Transformation:

\[
\hat{w} = w_0 + B\hat{\Theta}, \hat{\Sigma}_w = B\hat{\Sigma}B^T.
\]

where \( Tr \) denotes the trace of matrix and \( \Psi(\cdot) \) is the digamma function, respectively. After the posterior distribution over the parameters \( \Theta \) is obtained from the VB learning procedure, this distribution over \( \Theta \) must be transformed to the posterior distribution over the weights \( w \). Because the posterior distribution over \( \Theta \) is a Gaussian distribution and the transformation is linear, the posterior distribution over \( w \) is also a Gaussian distribution with mean \( \hat{w} = w_0 + B\hat{\Theta} \) and variance \( \hat{\Sigma}_w = B\hat{\Sigma}B^T \). The convergence of the algorithm can be measured by the increase of the lower bound \( F \) (Roberts and Penny, 2002).

2.3. Sequential algorithm for Bayesian GAFN

Another training strategy for Bayesian GAFN is to extract the relationships between inputs and output sequentially, i.e. one by one, rather than to learn all of them simultaneously. This results in the sequential algorithm for Bayesian GAFN. Fig. 3 illustrates a graphical model for sequential Bayesian GAFN. Total \( d \) latent variables \( z_i, (i = 1,...,d) \) are introduced into the graphical model to establish the probabilistic GAFN. The outputs of approximation functions are multiplied by the coefficients \( b_i, (i = 1,...,d) \) and the results serve as the inputs of hidden random variables \( z_i, (i = 1,...,d) \). After the introduction of hidden variables, which can also be expressed as a random vector \( Z \), we can get the following conditional distributions

\[
y \mid Z \sim \mathcal{N} \left( y \mid \hat{f}_d + 1 \left( \alpha^T Z \right), \psi_d \right)
\]

\[
z_i \mid x_i \sim \mathcal{N} \left( z_i \mid b_i \hat{f}_i \left( x_i \right), \psi_i \right), \quad i = 1, ..., d.
\]

where \( 1 \{ 1, \ldots, 1 \} \) is the unit vector, \( \psi_{d+1} \) is the noise variance of output and \( \psi_{2+1} \) are variances of hidden random variables \( z_i, (i = 1,...,d) \). In this work, \( y \) and \( z_i \) are used as random variables with normal distribution.

Following the strategy introduced in Section 2, we also place Gaussian prior over each coefficient \( b_i \) with mean 0 and precision \( \alpha_i \), and a gamma prior over each precision hyperparameter \( \alpha_i \). The mathematical expressions used in the work for the prior distributions are shown as follows

\[
p(b_i \mid \alpha_i) = \left( \frac{\alpha_i}{2\pi} \right)^{1/2} \exp \left\{ -\frac{c_i}{2} b_i^2 \right\}
\]

\[
p(\alpha_i) = \frac{d_i^e}{\Gamma(c_i)} \alpha_i^{c_i - 1} \exp(-d_i \alpha_i).
\]

where \( c_i \) and \( d_i \) are shape parameter and scale parameter respectively. If initial values of them are set to be zero, then the distributions
correspond to an infinitely broad prior over $\alpha_i$. This method is similar to the Automatic Relevance Determination (ARD) strategy for the learning of Bayesian Neural Networks (Neal, 1996). As one can see from above analysis, the variational Bayesian Backfitting algorithm can be applied for the learning of parameters of GAFN sequentially. According to Bayesian Backfitting algorithm (D’Souza et al., 2004; Vijayakumar et al., 2003), one can get an EM-like variational estimation method for the functional network model, as is shown in Algorithm 2.

**Algorithm 2.** Sequential learning for Bayesian GAFN

**Initialization:**

- $\tilde{f}, \tilde{c}_i, d_i^0$, tolerance, iteration, $i = 1, \ldots, d$;
- $\alpha_0^i = \alpha_i^0 + \frac{1}{2}$
- repeat
- **M-Step**
  
  $$(\hat{b}_i)^{t+1} = \sum_{t=1}^{N} \frac{\phi_i(x_{ni})}{\sum_{i=1}^{d} \phi_i(x_{ni})} + \frac{2}{\Psi + 1} \sum_{t=1}^{N} \left( x_{ni} - \sum_{i=1}^{d} (\hat{b}_i)^{t} \tilde{f}_i(x_{ni}) \right) \tilde{f}_i(x_{ni})$$

- **E-Step**
  
  $$(\hat{\psi}_i) = \left( \frac{1}{d} \sum_{i=1}^{d} \psi_i \right) \left( 1 - \frac{1}{2} \sum_{i=1}^{d} \psi_i \right)$$

- $$(\sigma^2_m) = \hat{b}_i \tilde{f}_i(x_m) + \frac{1}{2} \left( y_n - \sum_{i=1}^{d} \hat{b}_i \tilde{f}_i(x_m) \right)$$

- **Update the posterior distributions of coefficients $b_i$ and precision $\alpha_i$**

  $$(\alpha_i^2) = \left( \frac{1}{\Psi + 1} \sum_{i=1}^{d} \tilde{f}_i(x_{ni}) \right)^{-1}$$

- $$(\mu_i) = \frac{1}{\Psi + 1} \sum_{i=1}^{d} (\tilde{f}_i(x_{ni}) - (\alpha_i^2)^{-1})$$

- **Calculate $s = \sum_{i=1}^{d} \psi_i$ and update parameters in $\tilde{f}_i$**

end for until convergence of $\tilde{f}_i$ or maximum iterate.

3. Materials

3.1. Synthetic data

A specific synthetic data set is designed which may demonstrate better understanding about benefits of the proposed approach. This data set needs to be meaningful in principles to reflect dynamic characteristics of plant growth, such as both smooth nonlinearity or hard nonlinearity of functions. Suppose there are linear or nonlinear relationships between the response $Y$ and independent variables $x_i$, $i = 1, \ldots, d$, and a set of collected data $D = \{ (x_j, Y_i), j = 1, \ldots, N \}$, where $x$ is the independent variable vector. There are a lot of statistical techniques which may create models to fit the given data, such as parametric regression model and Neural Networks. It will be shown in this work that how Bayesian GAFN may play as an optimization tool to estimate some interested parameters on the basis of the given data set, and how Bayesian inference can be made on the obtained data starting with some special prior knowledge. In order to demonstrate the characteristic of Bayesian GAFN, a synthetic data is produced to illustrate that, under some particular conditions, the GAFN model can find out one optimal solution for an important parameter in nonlinear ecological systems. The synthetic data are generated from following functions, which are originated from the model introduced by Yin et al. (2003) to simulate the dynamic crop growth process

$$Y(S) = f(S(g(x(t))) + \varepsilon(t)$$

where

$$f(S(g(x))) = W \left( \frac{1 + \frac{t_{max} - S(g(x(t)))}{t_{opt} - t_{max}}}{\frac{t_{max} - t}{t_{opt} - t_{max}}} \right)$$

is beta function and $\varepsilon(t)$ is the noise part. Typically, $S(g(x(t)))$ is integral function with respect to variables $g(x(t))$ and $t$. The function $S(g(x(t)))$ has the following form

$$S(g(x(t))) = \int_0^t g(x(\tau))d\tau$$

where $g(x(t))$ is a nonlinear function having a hard threshold parameter $t_{th}$. If $x(t)$ is greater than $t_{th}$, the effective output is $x(t) - t_{th}$, and if $x(t)$ is smaller than $t_{th}$, then the output of function is $0$. That is

$$g(x(t)) = \begin{cases} x(t) - t_{th}, & \text{if } x(t) > t_{th}; \\ 0, & \text{if } x(t) < t_{th}. \end{cases}$$

This function is often used in agro-ecology for the description of the effect of temperature on crop (Mi et al., 2005). In order to simulate the evolitional characteristics of environmental factors in greenhouse, triangular functions are used to generate the data points of $x(t)$ (Kumari et al., 2007). The function is shown as following

$$x(t) = T_0 + A \sin(2 \pi \omega t) + \epsilon_f.$$  

(27)

where $T_0$ is the average bias of $x(t)$, $A$ is the fluctuation range, and $\omega$ is the frequency. The $\epsilon_f$ is the additive noise imposed on $x(t)$. In this paper, we are going to learn the preset parameter $T_{th}$ by Bayesian GAFN model from the given data set. It is observed that the data set generated from Eqs. (23)–(27) mimics a dynamic process of plant growth with hard nonlinearity of Eq. (26) and smooth nonlinearities from other equations.

3.2. Real data

The results of 12 experiments with indeterminately growing tomato (Lycopersicon esculentum ‘Counter’) were gathered from 1987 to 1992 in the greenhouse compartments of the Department of Horticulture in Wageningen, the Netherlands (Heuvelink, 1995). Several papers have been reported to analyze the data sets and evaluate models (Heuvelink, 1996, 1999). A total of 139 fresh and dry weight data were collected destructively on three to eight tomato plants every 6–22 days. In addition to the total weight, weight from leaves (including petioles), stem, individual fruit trusses, removed leaves and picked fruits and leaf area were also measured. However, in this paper, only the total dry weight is used as the output variable for simplicity. In addition to the botanical variables, several daily
Fig. 4. Cumulative dry matter production with respect to days from planting. The total data points are 139 and \( m \) represents the data points of each experiment. Data sets are adopted from Heuvelink (1995).

Fig. 5. (a) The function plot of \( x(t) \) with \( T_0 = 20, A = 5, \omega = 0.005 \) and \( \varepsilon = 0.3 \), respectively (b) The function plot of \( g(x(t)) \) with \( T_{th} = 16 \), (c) The function plot of \( S(g(x(t))) \), (d) The function plot of \( f(x(t)) \) with \( W = 100, T_{max} = 2000 \) and \( T_{opt} = 900 \), respectively.
environmental variables are also collected during the full growth period of tomato, for example, the daily global radiation, the daily temperature and the daily carbon dioxide concentration. For the limitation of space, please refer to Heuvelink’s papers for detail discussions about the experimental set-ups (Heuvelink, 1995). The 12 plots of cumulative dry weights with respect to days from planting are shown in Fig. 4. In Section 2, we will show how to make use of the Bayesian GAFN to model the actual plant growth process.

4. Experimental results

4.1. Synthetic examples

In this section, we apply the Bayesian GAFN to the estimation of interested parameter in a synthetic single-input and single-output (SISO) problem. There are two kinds of prior knowledge to be utilized in this problem, the threshold temperature $T_{th}$ in Eq. (26) and the integral with respect to temperature in Eq. (25), respectively. The data are generated from Eq. (23), where $\epsilon(t)$ is set to be zero so that there is no noise imposed on response. In Eq. (24), $W$, $T_{max}$, $T_{opt}$ are set to be 100, 2000 and 900 respectively. The input $x(t)$ is produced from Eq. (27) in terms of $T_{0}=20$, $A=5$, $\omega=0.005$ and $\epsilon_d=0.3$. The parameter to be estimated is $T_{th}$ in Eq. (26), which is set to be 16 in advance while generating the training and testing data. Totally 200 training data are generated by uniformly distributing $x(t)$ from 1 to 200, and the testing set contains 150 data points under the same condition. The function plots of Eqs. (24)–(27) are shown in Fig. 5.

The parallel algorithm is used for the estimation of parameter $T_{th}$ on the basis of the training data. For each given $T_{th}$, 20 different data sets are generated in random for the learning of Bayesian GAFN, and the variational lower bound is used as the criteria for choosing $T_{th}$. In comparison of the obtained lower bound, one can choose the most fitted value of $T_{th}$ from the obtained results. With respect to the network settings, the polynomial functionals are chosen as the approximation bases and the order obtained results. With respect to the problem in hand. It should be noted that function Eq. (24) is not differentiable in respect to $T_{th}$, so that the gradient-based methods are not suitable for solving this problem. However, Bayesian GAFN can provide an alternative approach for the estimation of interested parameter through a proper problem setting. Note that the convergence of Bayesian GAFN is very quick in comparison with other standard artificial intelligence techniques, such as neural networks and genetic algorithm. In general, only 4–5 s are required for the training procedure with respect to a given $T_{th}$. This example shows that the proposed approach is able to present not only good predictions on the dynamic nonlinear process (mean testing RMSE = 0.24 and mean training RMSE = 0.20), but also a reasonable estimation on $T_{th}$.

Fig. 7(a) - (c) demonstrate the evolution of the posterior distributions over weight precision $\alpha$ and noise precision $\beta$. In general, one has little prior information about the distribution of the weights and noise. Therefore, under the evidence framework, we often consider broad priors over the precision parameters. Note that the inverse of precision is the variance of distribution, so the precision parameter is positive and is often assumed to have Gamma distribution. The smaller value of posterior precision means a broad distributional variance. For example, the prior mean of the parameters $\theta$ is assumed to be zero. If the posterior mean of $\theta$ is also presumed to be around zero, then the smaller precision means that the posterior estimations significantly deviate from zero, and they should not be supposed to be zero anymore in a sense. This is the essence of the Automated Relevance Determination (ARD) strategy and it can be used for model selection procedure. This method is introduced by R.M. Neal in 1996 for the Bayesian learning of neural networks. As one can see from these figures, the distributions of $\alpha$ and $\beta$ shrink from broad priors to narrow posteriors after the computation. The posterior mean of weight precision $\alpha$ is very close to zero, which means that the estimations of model parameters $\theta$ significantly deviate from zero, and there is no redundant parameters in current model. Note that, as is shown in Fig. 7(d), the variational procedure provides a consistent and monotonic increasing approximate lower bound for the log marginal probability, and the lower bound converges to the upper bound after only 3–5 steps in 1–2 s.

4.2. Real-world examples

In this section, Bayesian GAFNs are applied to a practical problem: modeling dynamic greenhouse crop growing process. We have introduced the twelve real-world greenhouse tomato experiments in Section 2, and have indicated that there are totally 139 data points collected. In our model, five environmental factors are chosen as the

![Figure 6](image_url)
independent variables, and one crop characteristic serves as the dependent variable, which are

\[ x_1 \] the planting date,
\[ x_2 \] the weight at planting date,
\[ x_3 \] the accumulated global radiation,
\[ x_4 \] the temperature degree day,
\[ x_5 \] the total carbon dioxide concentration,
\[ y \] the total growth dry weight.

respectively. First of all, we have a hypothesis that the relationships between independent variables conform to associativity, that is, any permutation of the \( x_1, x_2, x_3, x_4, x_5 \) will produce the same argument of \( y \).

For the limit of paper space, please refer to Aczél (1966) and Castillo et al. (1999) for a detailed discussion. Under the associativity hypothesis, Bayesian GAFNs could be applied to the modeling of crop growth process.

In this section, parallel algorithm is first applied to the modeling of dynamic growth data, and three statistical analysis: model selection, model diagnostic and quantification of variable contribution, are provided for model validation. Then the performance of Bayesian GAFN models is compared with that of linear regression model and neural networks under Bayesian frameworks based on the given data set.

### 4.2.1. Model selection

As we have mentioned above, the variational lower bound can serve as a criteria for model selection. As the number of samples increases, the parameter posterior becomes sharply peaked around the most probable values, and in the large sample limit \( N \rightarrow \infty \), the variational lower bound becomes equivalent to the Bayesian information criterion (BIC) (Attias, 2000; Roberts and Penny, 2002). In Bayesian formulation, all values of parameters are integrated out through marginalizing the joint probability of data and model. First of all, we need select an optimal order of the polynomial neuron functions to give the best generalization performance. Twelve experimental data sets are divided into training set and testing set circularly. Eleven data

![Fig. 7.](https://example.com/fig7.png)

**Fig. 7.** (a) The prior distributions for precision parameter \( \alpha \) and \( \beta \), (b) The posterior distribution of \( \alpha \), (c) The posterior distribution of \( \beta \). (d) The evolution of variational free energy.

<table>
<thead>
<tr>
<th>No.</th>
<th>Model order</th>
<th>Num. of para.</th>
<th>Variational lower bound</th>
<th>BIC Training</th>
<th>RMSE Training</th>
<th>RMSE Testing</th>
<th>RMSE Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[2 2 2 2 2]</td>
<td>6</td>
<td>-961.30</td>
<td>1073.69</td>
<td>59.05</td>
<td>72.45</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>[2 2 2 2 3]</td>
<td>7</td>
<td>-1049.61</td>
<td>1077.97</td>
<td>58.69</td>
<td>76.20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>[2 2 2 2 3]</td>
<td>7</td>
<td>-1042.37</td>
<td>1060.50</td>
<td>54.79</td>
<td>68.84</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>[2 2 3 2 2]</td>
<td>7</td>
<td>-1025.75</td>
<td>1027.41</td>
<td>48.11</td>
<td>59.13</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>[2 3 2 2 2]</td>
<td>7</td>
<td>-1048.81</td>
<td>1076.73</td>
<td>58.40</td>
<td>96.48</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>[3 2 2 2 2]</td>
<td>7</td>
<td>-1039.22</td>
<td>1048.32</td>
<td>52.22</td>
<td>69.13</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>[3 3 2 2 2]</td>
<td>8</td>
<td>-1131.91</td>
<td>1032.99</td>
<td>48.06</td>
<td>53.73</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>[2 3 3 2 2]</td>
<td>8</td>
<td>-1124.80</td>
<td>1029.06</td>
<td>47.31</td>
<td>74.70</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>[2 2 3 3 2]</td>
<td>8</td>
<td>-1124.82</td>
<td>1029.95</td>
<td>47.48</td>
<td>58.89</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>[2 2 2 3 3]</td>
<td>8</td>
<td>-1140.28</td>
<td>1053.98</td>
<td>52.18</td>
<td>68.44</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>[2 2 3 2 3]</td>
<td>8</td>
<td>-1126.71</td>
<td>1031.40</td>
<td>47.75</td>
<td>60.86</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>[3 2 2 3 2]</td>
<td>8</td>
<td>-1142.32</td>
<td>1062.76</td>
<td>54.03</td>
<td>96.46</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>[3 3 2 2 2]</td>
<td>8</td>
<td>-1125.55</td>
<td>1025.58</td>
<td>46.67</td>
<td>61.50</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>[3 2 3 2 3]</td>
<td>8</td>
<td>-1134.37</td>
<td>1037.81</td>
<td>48.96</td>
<td>67.64</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>[3 2 3 2 2]</td>
<td>8</td>
<td>-1149.54</td>
<td>1080.99</td>
<td>58.04</td>
<td>99.78</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>[3 2 3 3 2]</td>
<td>8</td>
<td>-1139.39</td>
<td>1045.98</td>
<td>50.56</td>
<td>70.16</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>[3 2 3 2 2]</td>
<td>9</td>
<td>-1132.02</td>
<td>1035.15</td>
<td>47.35</td>
<td>61.20</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>[3 3 2 2 2]</td>
<td>9</td>
<td>-1234.77</td>
<td>1031.60</td>
<td>46.69</td>
<td>66.91</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>[3 2 3 2 2]</td>
<td>9</td>
<td>-1235.77</td>
<td>1031.63</td>
<td>43.39</td>
<td>54.95</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>[3 2 2 3 3]</td>
<td>9</td>
<td>-1238.87</td>
<td>1021.14</td>
<td>44.84</td>
<td>58.71</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>[3 2 3 2 2]</td>
<td>9</td>
<td>-1240.34</td>
<td>1031.32</td>
<td>46.64</td>
<td>62.10</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>[3 2 3 2 2]</td>
<td>9</td>
<td>-1252.31</td>
<td>1054.79</td>
<td>51.14</td>
<td>93.15</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>[3 3 2 2 2]</td>
<td>9</td>
<td>-1237.65</td>
<td>1032.63</td>
<td>46.88</td>
<td>72.18</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>[3 3 2 2 2]</td>
<td>9</td>
<td>-1238.76</td>
<td>1030.97</td>
<td>46.58</td>
<td>60.99</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>[3 2 2 3 3]</td>
<td>9</td>
<td>-1249.22</td>
<td>1042.96</td>
<td>48.82</td>
<td>68.52</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>[3 3 2 2 2]</td>
<td>9</td>
<td>-1243.73</td>
<td>1027.26</td>
<td>45.93</td>
<td>54.48</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>[3 2 3 3 3]</td>
<td>10</td>
<td>-1361.22</td>
<td>1036.98</td>
<td>46.59</td>
<td>70.05</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>[3 2 2 3 3]</td>
<td>10</td>
<td>-1364.91</td>
<td>1035.49</td>
<td>46.33</td>
<td>62.50</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>[3 3 2 2 3]</td>
<td>10</td>
<td>-1366.50</td>
<td>1026.72</td>
<td>44.78</td>
<td>62.32</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>[3 3 2 2 3]</td>
<td>10</td>
<td>-1360.60</td>
<td>1020.81</td>
<td>43.75</td>
<td>57.07</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>[3 3 3 2 2]</td>
<td>10</td>
<td>-1359.74</td>
<td>1021.73</td>
<td>43.91</td>
<td>55.44</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>[3 3 3 2 3]</td>
<td>11</td>
<td>-1498.59</td>
<td>1026.23</td>
<td>43.67</td>
<td>56.33</td>
<td></td>
</tr>
</tbody>
</table>
sets are fallen into training set, and one data set into testing set. Therefore, in total twelve training errors and twelve testing errors are obtained during the learning procedure. Then the results are averaged to produce the mean errors for performance comparison.

Bayesian frameworks prevent the overfitting problem by integrating out the parameters following the predictive approach. So one can compare two models directly by the ratio of their evidence, known as the Bayes factor. In the formulation of variational framework, the lower bound can be used for model comparison instead. It is worth noting that the model order of GAFN is a discrete combination of some positive integers. In definition, there are infinite number of models to be considered by combination of infinite integers. However, only order 2 and 3 are considered with respect to each approximation functionals for the current problem. This treatment is reasonable in that the second-order polynomial function is enough to explain this problem because of the small sample data set. For five input variables, there are totally $2^5 = 32$ possible models for greenhouse crop growth data. We take the prior parameters $c_0 = d_0 = 0.001$, $c_1 = d_1 = 0.001$, $\Delta F = 1 \times 10^{-5}$ respectively. Table 1 gives the lower bound and BIC with respect to different model. The value of BIC is calculated by the following equation (Venables and Ripley, 2002)

$$BIC = N \log \left( \sum_{i=1}^{N} \left( \frac{y_i - \hat{y}_i}{N} \right)^2 \right) + p \log(N)$$  \hspace{1cm} (28)

where $N$ is the number of data points, $y_i$ is the observation, $\hat{y}_i$ is the output of Bayesian GAFN and $p$ is the number of model parameters.

As one can see from Table 1 and Fig. 8, the variational lower bound criterion penalized the complex model, and prefer the model with least parameters. The model 1 with greatest variational lower bound do not provide the least testing RMSE. Model 19 with least BIC has a testing error closer to the lowest testing error. As can be see from Fig. 8, the variational lower bound is sensitive to the number of parameters, that is, the model complexity. Note that, in this practical application, variational lower bound only favored the most simple model and underfit the given data. It appears that the model under the BIC criterion is more suitable for the this problem because its generalization performance ranks higher among the obtained models. Although the variational method formed a strict lower bound for the marginal log-likelihood, the variational approximation would get worse with increasing numbers of parameters. This is to be expected due to the factorized nature of the variational approximation method. The factorization omits the correlation of the parameters, and it will produce much more error while the number of parameters increasing. Fig. 9 gives a plot of the regression curves for the 139 data points from the model under BIC criterion.

4.2.2. Model diagnostics

Regression model building is often an iterative and interactive process, and model diagnostics are needed to detect problems with the model and suggest improvements. Checking the outliers and normality assumptions is the most often-used diagnostic techniques. Firstly, we
need to check the normality assumption of the model through the observations of residuals. The residual is defined as the difference between the observed value $y_i$ and the fitted value $\hat{y}_i$, which can be written as $\varepsilon_i = y_i - \hat{y}_i$. Normally, one would also check the absolute residual versus the fitted value. So we plot the residuals $\varepsilon$s and the absolute residuals |$\varepsilon$|s against the fitted values $\hat{y}$s in Fig. 10(a) and (b), respectively. One can observe that the variances show great heteroscedasticity and become more broad with the increase of the fitted values. That is, while the growth of crop proceeding, the biomass production will show more uncertainty than the primary phase and the prediction of crop growth will become more difficult. This result is reasonable according to the biological knowledge. For example, as the tomato grows, there would be more leaves on the stems, and the light interception condition would become more complicate in comparison with that of seedling phase. Then the growth tendency of crop would show more diversity and the production of total dry weight would be different as well, even in the same greenhouse.

One can check the normality assumption and outliers from the $Q$–$Q$ plot and boxplot. Fig. 10(c) is the $Q$–$Q$ plot of quantile of samples in comparison with the quantile of standard normal distribution. As one can see from the figure, the $Q$–$Q$ plot shows a kind of non-constant normality because there shows large deviation between the points and the straight line at the end of tails. However, if one examines the plot carefully, only three out of 139 data points are deviate significantly from the straight line. In order to locate the outliers in the sample, one can plot the boxplot of residuals, as is shown in Fig. 10(d). According to the boxplot, one can find three outliers effortlessly, two in the upper side and one in the lower side. By excluding these three
points, the normality will become a reasonable assumption for the plant growth problem. Fig. 10(e) and (f) give the Q–Q plot and the histogram of residuals respectively. The normality becomes apparent after the deletion of three outliers. The diagnostic analysis makes it clear that the assumption of normality is suitable for the crop growth model. A close observation of outliers may also provide some useful knowledge for one to understand the plant growth process. One can see that the three outliers are located near the middle range of total dry weight of tomato growth, which are near 600–800 (g/m²). It is well known that a plant or a crop growth duration can be divided into three sub-phases: an early accelerating phase; a linear phase; and a saturation phase for ripening. Therefore, the growth pattern typically follows a sigmoid curve, and the growth rate a bell-shaped curve (Yin et al., 2003). So, in the middle of growth phase, the rate of growth will reach its maximum, and the crop will show its greatest diversity during this period. The analysis of outliers conforms to the prior biological knowledge about three sub-phase of growth.

4.2.3. Quantification of variable contribution

Next Bayesian GAFN approach is applied to quantifying the variable contributions of every inputs for the output. There are a lot of statistical methods for quantifying variable importance, for example the entropy or correlation-based feature selection strategies (Xing and Hu, 2009; Peng et al., 2005; Torkkola, 2005; Guyon and Elisseeff, 2003). Some typical strategies include connection weights, Garson’s algorithm, input perturbation, sensitivity analysis, forward stepwise addition and backward stepwise elimination, etc. (Olden et al., 2004). The two most simple methods are forward stepwise addition and backward stepwise elimination. Stepwise procedures are relatively cheap computationally but they tend to pick models that are smaller than desirable for prediction purpose. In this work, we will use forward stepwise method to quantify the variable contribution of inputs. In order to use the results from Section 1, we started with no variables in the model and continue until no new predictors can be added into the Bayesian GAFN. For every possible combination of predictors, we will compute their BICs with order 2 or 3 and choose the lowest BIC for comparison. For the greenhouse growth problem, there are totally \(C_5^1 \times 2^1 + C_5^2 \times 2^2 + C_5^3 \times 2^3 + C_5^4 \times 2^4 + C_5^5 \times 2^5 = 242\) possible cases. Fig. 11 gives a plot of the best BICs for every possible variable-combination models. The models are denoted by the indices for predictors.

As one can see from the figure, the variable combination of \((x_1, x_2, x_3, x_5)\) produces the best result under the BIC criterion. The model \((x_3, x_2, x_3)\) is rather comparable to the two four-predictor models \((x_1, x_2, x_3, x_4)\) and \((x_1, x_2, x_3, x_5)\), and the full model \((x_1, x_2, x_3, x_4, x_5)\). As far as the individual predictor is concerned, \(x_3\), the global radiation, has the most influential effect on the growth process. The next two factors are temperature \(x_4\) and carbon dioxide concentration \(x_5\). The first two independent variables \(x_1\), the planting date, and \(x_2\), the weight at planting date, have the least effects on the plant growth process. Obviously, the global radiation \(x_3\) is the most important environmental factor that affects the crop growth, because it appears in all the models which have the best performance in terms of the number of predictors. This result conforms to biological experience that photosynthesis is the fundamental function of plant growth process. Although the planting date and the weight at planting date are not very important to the plant growth prediction individually, it would be greatly helpful to combine other independent variables with them for a better prediction accuracy. Some interesting phenomena occur in the BIC plot of greenhouse crop growth. Variables \(x_4\) and \(x_5\) are significantly better than \(x_1\) and \(x_2\) in terms of BIC, one would expect 345 should perform better than 123, unless there are strong correlations among \(x_1, x_4, x_5\). Shorter combinations such as 13 also perform better than 35/34. On the other hand, ‘12’ scores rather poor, but somehow ‘123’ becomes the best in 3-variable combinations. In fact, the global radiation \(x_3\) has a great effect on the crop growth process. As one can see from Fig. 9, the crops grow slowly in winter because of the less radiation. The \(x_5\) planting dates, can be looked as the seeding season of crop. The effects of \(x_1\) on crop growth is magnified with the combination with the global radiation. As far as the variables \(x_4\) and \(x_5\) are concerned, the temperature \(x_4\) and the carbon dioxide concentration \(x_5\) in the greenhouse do not differ very much during a day, and their effects on the dynamic growth process are weakened by the lack of data variance. Furthermore, there are threshold values of \(x_4\) and \(x_5\) which have important effects on the plant growth process, and make the relationships between \(x_4, x_5\) and output non-linear. For example, while the greenhouse temperature under the threshold, the plant would slow or stop growth process because of the low temperature. While the carbon dioxide concentration is greater than the threshold, superfluous carbon dioxide will have no effect on the dynamic plant growth process. However, no temperature under threshold and carbon dioxide up the threshold are collected in the course of the 12 experiments. So the deficit of data collection makes the results not fully reflect the importance of the combination of \(x_4\) and \(x_5\) with \(x_3\) as well.

4.2.4. Sequential learning for greenhouse data

We have introduced that GAFN model can also be learned by Bayesian backfitting algorithm. The sequential algorithm is different from the parallel algorithm in terms of learning strategy. The sequential algorithm learns the input–output relationship alternately and the neurons are optimized respectively. Unlike the parallel learning method, there are no constraints on model parameters anymore. So the number of free parameters in sequential strategy is greater than that of parallel algorithm with respect to same model structure. For example, if the model order is [33322], then there are 9 free model parameters in parallel algorithm, and 13 free model parameters for sequential algorithm in contrast. However the approximate performance would almost keep the same as that of parallel strategy, as shown below. With respect to the initial conditions, the prior hyperparameters for coefficients \(b_i\) take \(c_i^0 = d_i^0 = 0.001\), initial value for \(b_i\) takes \(1 \ (i = 1, \ldots, d)\), and the maximum iteration takes 1000, respectively. As the model setting in Section 1, we explored the performance of sequential GAFN on all the 32 models. Variational lower bound, BIC, training RMSE and testing RMSE are calculated for model comparison. The best model selected by the BIC criteria is [33322] as the parallel algorithm does, where lower bound is = 1253.5497, BIC is 1032.03, training RMSE is 45.58 (g/m²) and testing RMSE is 52.11 (g/m²) respectively. It is worthy noting that BIC of sequential algorithm is greater than that of parallel algorithm and the lower bound of parallel one is lower than that of sequential one, because there are no constraints on the parameters in the backfitting strategy anymore, and the sequential algorithm has more parameters than the parallel algorithm with respect to the same model. Both methods chose the same model.
The estimates from three different algorithms with respect to GAFN and linear models on greenhouse crop data (the model order is $M = [3 \ 3 \ 2 \ 2]$) over whole data set.

<table>
<thead>
<tr>
<th>Weights</th>
<th>Bases</th>
<th>Seq. GAFN</th>
<th>Para. GAFN</th>
<th>Linear regression</th>
<th>Neural networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_{11}$</td>
<td>1</td>
<td>-138.09</td>
<td>-4.44</td>
<td>302.54</td>
<td></td>
</tr>
<tr>
<td>$w_{12}$</td>
<td>$x$</td>
<td>-48.30</td>
<td>-35.37</td>
<td>-35.74</td>
<td></td>
</tr>
<tr>
<td>$w_{13}$</td>
<td>$x^2$</td>
<td>50.22</td>
<td>40.81</td>
<td>42.41</td>
<td></td>
</tr>
<tr>
<td>$w_{14}$</td>
<td>1</td>
<td>-97.92</td>
<td>1.29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$w_{21}$</td>
<td>$x$</td>
<td>20.99</td>
<td>17.91</td>
<td>17.83</td>
<td></td>
</tr>
<tr>
<td>$w_{22}$</td>
<td>$x^2$</td>
<td>-20.30</td>
<td>-17.90</td>
<td>-17.94</td>
<td></td>
</tr>
<tr>
<td>$w_{23}$</td>
<td>1</td>
<td>288.78</td>
<td>-288.43</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>$w_{24}$</td>
<td>$x$</td>
<td>211.86</td>
<td>262.87</td>
<td>261.13</td>
<td></td>
</tr>
<tr>
<td>$w_{31}$</td>
<td>$x$</td>
<td>23.71</td>
<td>26.56</td>
<td>26.40</td>
<td></td>
</tr>
<tr>
<td>$w_{32}$</td>
<td>1</td>
<td>203.12</td>
<td>21.79</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>$w_{33}$</td>
<td>$x$</td>
<td>65.15</td>
<td>-20.79</td>
<td>-14.35</td>
<td></td>
</tr>
<tr>
<td>$w_{34}$</td>
<td>1</td>
<td>44.16</td>
<td>573.80</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>$w_{35}$</td>
<td>$x$</td>
<td>-12.27</td>
<td>43.14</td>
<td>37.83</td>
<td></td>
</tr>
</tbody>
</table>

Para. | 13 | 9 | 9 | 15 | RMSE (g/m$^2$) | 45.16 | 44.42 | 44.43 | 46.64 |

According to the BIC criterion, the T-test shows that the sequential method has equal training and testing performance comparing with the parallel one. The mean estimates of $b_i (i = 1, 2, 3, 4, 5)$ are $b_1 = 0.8299$, $b_2 = 0.8114$, $b_3 = 0.9096$, $b_4 = 0.9434$ and $b_5 = 0.9765$, respectively. The estimates of $b_i$ are coupled with the function parameters in $f_i(x_i)$. Note that the two algorithms obtained different estimates of neuron parameters even under same model structure. With respect to the polynomial neuron functions, GAFN model can be looked as a generalization of linear polynomial model. If linear model is written as $y = \alpha + \beta X$, the bias items $\alpha$ can be regarded as the sum of the bias in neuron function $f_i(x_i)$, which represents the mean effect of response. The generalized associativity functional equation gives an answer to the question that under what conditions the additive model can be used for model building: if the variables conform to associativity, then the sum can be used for the associative operation between the approximate functions of variables.

In order to compare the performance between Bayesian GAFN and other statistical models, we illustrate the estimates of errors over whole data set, which are obtained from parallel algorithm, sequential algorithm of GAFN. In addition, linear model and neural networks are also used for the performance comparison, which are learned under the Bayesian framework. Table 2 gives the results from the four algorithms. The learning method for neural networks is also a neural model under the Bayesian framework (Bishop, 1995). The model for neural networks is selected according to the same model choose strategy described in Section 1. The computation results show that 3-layer neural networks with 2 hidden neurons has the best testing RMSE, 64.97 (g/m$^2$), and the training RMSE is 45.89 (g/m$^2$). If this neural networks model is applied to the whole data set, it gives the prediction RMSE 46.64 (g/m$^2$), as is shown in Table 2.

The number of independent parameters in linear model is smaller than that of both GAFN models. However, the number of independent model parameters in parallel GAFN is identical to that of linear regression model. As one can see from the results, the approximate performance of parallel algorithm is comparable to linear regression model, and the three bias items are near to each other. However, the estimates of model parameters are not equal to those of others exactly, and some of them even have opposite signs. This is because of the uncertainty property of Bayesian framework, and is due to the learning strategy of different methods.

5. Discussion

Based on the functional equations, we have proposed variational Bayes learning procedures for the Generalized Associative Functional Networks — GAFN model. Firstly, the VB framework provides an efficient approximation of lower bound of marginal likelihood and posterior distributions over parameters, not point estimations of parameters. Although other methods, like bootstrap or Markov Chain Monte Carlo, may also provide an approximation of posterior distributions of model parameters, they are considered computationally intensive and the assessment of their convergence is often problematic (Roberts and Penny, 2002). After the posterior distribution is obtained by the VB method, statistical inference over the model size and model parameters can be drawn conveniently and the predictions will be made by integrating the distribution of parameters. Secondly, in the implementation of applying a variational Bayesian framework on parallel GAFN learning strategy, one needs to impose equality constraints on model parameters for reaching a unique solution of parameters. Instead of resorting to the Lagrange multiplier method, we describe a simple algorithm that linearly transforms the weights from original space to a subspace in terms of the equality restrictions. Therefore, the number of independent parameters is the number of original total weights minus the number of constraints. Although VB learning is intrinsically iterative and local optimal, the experimental results are acceptable in terms of the computational cost and predictive accuracy in comparison with other existing methods.

As far as the applications are concerned, one can see that the variational lower bound can be used as a criteria for the estimation of interested model parameter. As is shown in the synthetic experiment, Bayesian GAFN may accurately estimate model parameter according to lower bound criterion as long as the problem is properly set, even if a non-differentiable question. In fact the Bayesian GAFNs can even deal with more complicated problems at the cost of estimation accuracy. Moreover, while GAFN model is applied to the modeling of dynamic crop growth process, we have analyzed its statistical characteristics under the Bayesian framework. The experiment results have shown three main advantages of the Bayesian GAFN methods. First, Bayesian GAFNs may adapt to the inputs of multivariate system. That is, if one needs to study the effect of an additional independent variable on dependent variable, this input can be included into the model directly, just as what the neural networks have done. Second, Bayesian GAFN may serve as a useful data mining tool to explore functional relationships between inputs and outputs from the obtained data. By applying Bayesian GAFN to greenhouse crop data, the revealed underlying physical relations are consistent with the prior agro-ecological knowledge, and the ranking and combination of predictors are helpful for us to understand the plant growth process. Finally, Bayesian GAFNs present a good generalization capability from the final single model for successfully fitting all 12 data sets over 5 years field experiments. Although the 12 data sets were collected in different times and under different environmental conditions, the model could fit the whole data points accurately as long as enough environmental factors were considered with respect to the built model.

Another interest of this work is focused on the incorporation of prior knowledge in the modeling process. The nature world is neither "totally white" nor "totally black", indicating that not all models should be built to a "completely black box" (Hu et al., 2007). Whenever prior information is available, it should be included in models to represent that knowledge about the world explicitly. The GAFN is also an attempt to incorporate prior knowledge into the model building. With the application to the dynamic crop growth process, the associativity of input variables is used to simplify the network structure. The experimental results show that the performance is rather comparable to the classical neural networks with less parameter. From an application viewpoint, we need an applicable and simple modeling tool that is able to incorporate any form of prior information. Bayesian framework is also a good tool for the incorporation and extraction of knowledge from the given data. In this work, we also make an attempt to estimate the threshold temperature from a synthetic example. The
(results show that GAFN model can be used as an approximation tool for some interesting parameters. Although the simulation results are promising, there are still several important issues that need to be studied in future works. First, only linear equality constraints are investigated for the GAFN model in this work. However, there may be nonlinear approximation basis functions for FN, especially when the functional forms with respect to some variables are known as nonlinear functions. If the nonlinear basis functions appear in the GAFN model, the transformation of weights may become complicated, so does the transformation of the posterior distributions. Second, how to apply the VB framework to other types of FN models is also left to be investigated in future studies. For other types of FN models, the constraints may also become nonlinear functions with respect to model parameters. In this case, Lagrange multiplier methods may be a good choice. Finally, although the Bayesian framework has a built-in function of “Occam’s Razor”: Bayesian theorem automatically penalizes complicated models, assigning them large probability only if the complexity of the data justifies the additional complexity of the model (Gregory, 2005), VB framework also shows underfitting characteristics when applied to the greenhouse crop growth data. How to analyze and prevent this phenomenon needs more work to be done not only in mathematical fields but also in various engineering fields.

References


