Neural Comput & Applic
DOI 10.1007/s00521-014-1710-1

Variational Bayesian extreme learning machine

Yarui Chen · Jucheng Yang · Chao Wang ·
DongSun Park

Received: 13 September 2013 / Accepted: 9 September 2014

Abstract Extreme learning machine (ELM) randomly generates parameters of hidden nodes and then analytically determines the output weights with fast learning speed. The ill-posed problem of parameter matrix of hidden nodes directly causes unstable performance, and the automatical selection problem of the hidden nodes is critical to holding the high efficiency of ELM. Focusing on the ill-posed problem and the automatical selection problem of the hidden nodes, this paper proposes the variational Bayesian extreme learning machine (VBELM). First, the Bayesian probabilistic model is involved into ELM, where the Bayesian prior distribution can avoid the ill-posed problem of hidden node matrix. Then, the variational approximation inference is employed in the Bayesian model to compute the posterior distribution and the independent variational hyperparameters approximately, which can be used to select the hidden nodes automatically. Theoretical analysis and experimental results elucidate that VBELM has stabler performance with more compact architectures, which presents probabilistic predictions comparison with traditional point predictions, and it also provides the hyperparameter criterion for hidden node selection.

Keywords Extreme learning machine · Variational approximation · Bayesian model · Probabilistic prediction

Y. Chen · J. Yang (✉) · C. Wang
School of Computer Science and Information Technology,
Tianjin University of Science and Technology,
Tianjin 300222, China
e-mail: jcyang@tust.edu.cn

D. Park
Department of Electronic and Information Engineering,
Chonbuk National University, Jeonju,
Jeonbuk 561756, Republic of Korea

1 Introduction

Feedforward neural networks have wide application in many fields for its strong representation ability. The traditional gradient-based learning algorithms on the networks are usually slower than required. Huang et al. have proposed a novel learning algorithm with extremely fast learning speed, named extreme learning machine (ELM), which assigns the parameters of the hidden nodes randomly and analytically determines the weights linking the hidden layer and the output layer based on the Moore–Penrose generalized inverse of hidden layer output matrix $\mathbf{\Phi}$ [1, 2]. For its extremely fast learning speed and concise mathematical knowledge, the algorithm has been successfully applied in many areas, such as classification [3, 4], regression [5–7], pattern recognition [8, 9], and protein structure prediction [10].

The key problem of ELM is to compute the Moore–Penrose generalized inverse $(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}$ of matrix $\mathbf{\Phi}$. When the generalized inverse of matrix is ill-posed, the output weights $\mathbf{w}$ tend to large values, which cause weaker stabler and lower prediction accuracy [7]. Besides, experimental studies have shown that the ELM performance is stable in a wide range of number of hidden nodes, and ELM is not very sensitive to the number of hidden nodes. The selection of the stable number of hidden nodes is an important problem for ELM. The manual selection is unreliable for large-scale models and practical application, and then, automatical selection method is desirable. Huang et al. [11] propose an enhanced method for incremental extreme learning machine to select the hidden nodes automatically, where the hidden nodes generated randomly with the largest residual error will be added to the existing network at each learning step. But the method presents huge network architectures comparison with the traditional cross-validation method.

Published online: 24 September 2014

Variational method is a determined approximate inference method, which transforms the problem of probabilistic inference into the problem of functional optimization. The variational approximations originated from the free energy of statistical physics, such as the mean field free energy, Bethe/Kikuchi free energies [12–14]. Then, Wainwright and Jordan proposed a parameterized variational inference framework based on the convex duality theory in the graphical models with exponential distributions [15], which covered all of the standard variational inference methods, and also derived many new methods [16]. Variational methods also were used to approximate the Bayesian learning, where the variable integrals are involved for the Bayesian parameters and hidden variables, and the exact inference was intractable [17–19]. The variational approximate methods have become the favorite of approximate inference community due to its sound theoretical foundation and high convergence rate [20, 21].

We bring the Bayesian framework and variational approximate to ELM, referred to as variational Bayesian extreme learning machine (VBELM). In the VBELM model, the problem of learning the output weight **w** can be transformed into the problem of probabilistic inference about the Bayesian parameter variable, where the exact inference is intractable, and variational inference is adopted. The VBELM algorithm can improve the unstable performance of ELM by adding Bayesian prior knowledge and also select the hidden node number automatically using the variational independent hyperparameters.

In this paper, we first design the VBELM model through introducing the Bayesian prior distribution. Then, we execute variational approximate inference in the model to compute the posterior distribution of the target parameter and the model hyperparameters. Finally, execute the probabilistic prediction based on the VBELM algorithm. The advantages of the VBELM include: (1) generalization performance using the prior distribution of the Bayesian framework, (2) probabilistic predictions comparison of traditional point predictions, (3) automatic selection of hidden nodes through the hyperparameters.

The remainder of this paper is organized as follows. Section 2 introduces the ELM algorithm and the variational method. In the Sect. 3, we make a detailed description of VBELM, including the VBELM modeling, the corresponding variational inference procedure, the hidden node selection, and the probabilistic prediction based on the VBELM. Section 4 presents numerical experiments to show the efficiency of the VBELM in both the synthetic data sets and the standard data sets, and it also provides experiments to hidden node selection with hyperparameter criterion. Finally, Sect. 5 summarizes our work with some considerations on future directions.

2 Preliminaries

In this section, we introduce the ELM and the variational method.

2.1 Extreme learning machine

ELM is a fast learning algorithm for single-hidden layer feedforward neural networks (SLFNs), which randomly choose input weights of hidden nodes and analytically determines the output weights of SLFNs. Let $\{\mathbf{x}_i, t_i\}_{i=1}^{N}$ denote the samplers with input vector $\mathbf{x}_i = [x_{1i}, \ldots, x_{ni}]^{\mathbf{T}}$ and output t_i, N the sample number, n the dimension of input vector, M the number of the hidden nodes, $\phi(\cdot)$ the activation function, the mathematic model of the ELM is [1, 2]:

$$\sum_{j=1}^{M} w_i \phi_i(\mathbf{c}_j \mathbf{x}_i + d_j) = t_i, \quad i = 1, \ldots, N, \tag{1}$$

where $\mathbf{c}_j = [c_{j1}, \ldots, c_{jn}]^{\mathrm{T}}$ is the parameter vector connecting the jth hidden node and the input nodes, d_j is the threshold of the jth hidden node, and w_j is the weight connecting the jth hidden node and the output node.

The above N equations can be written compactly as:

$$\boldsymbol{\Phi}\mathbf{w} = \mathbf{t}. \tag{2}$$

where

$$\Phi(\mathbf{c}_1, \ldots, \mathbf{c}_M, d_1, \ldots, d_M, \mathbf{x}_1, \ldots, \mathbf{x}_N) = \begin{bmatrix} \phi_1(c_1 x_1 + d_1) & \cdots & \phi_M(c_M x_1 + d_M) \\ \vdots & \ddots & \vdots \\ \phi_1(c_1 x_N + d_1) & \cdots & \phi_M(c_M x_N + d_M) \end{bmatrix},$$

$$\mathbf{w} = [w_1 \ldots w_M]^{\mathrm{T}}, \quad \mathbf{t} = [t_1 \ldots t_N]^{\mathrm{T}}.$$

The ELM procedure involves the follow two steps:

1. Assign randomly hidden node parameters $\{\mathbf{c}_j, d_j\}_{j=1}^{M}$ from any intervals of R^n and R.
2. Calculate the output weights $\mathbf{w}$ by $\mathbf{w} = (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}$, where the quantity $\boldsymbol{\Phi}^{\dagger} \equiv (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}$ is known as the Moor–Penrose pseudo-inverse of the matrix $\boldsymbol{\Phi}$.

2.2 Variational method

For a model $p(\mathbf{x}, \mathbf{z})$, where $\mathbf{x} = \{x_1, \ldots, x_N\}$ is the set of all observed variables, $\mathbf{z} = \{z_1, \ldots, z_M\}$ is the set of all latent variables and parameters. The inference goal is to compute the posterior distribution $p(\mathbf{z}|\mathbf{x})$ and the model evidence $p(\mathbf{x})$. Since exact inference is intractable for large-scale model, various approximate inference methods are proposed. The variational inference is a determined approximate method, which transforms the inference problem of variable summing into a optimization problem of free distribution and computes the approximate posterior distribution through solving the optimization [15, 22].

According to the variational framework, the problem of computing the marginal probability $p(\mathbf{x})$ can be transformed into the optimization problem of the free distribution $q(\mathbf{z})$, that is

$$\begin{aligned}\ln p(x) &\geq \sup_{q(\mathbf{z})} \int q(\mathbf{z}) \ln \frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})} \mathrm{d}\mathbf{z} \\ &= \sup_{q(\mathbf{z})} \left\{ \int q(\mathbf{z}) \ln p(\mathbf{x},\mathbf{z}) \mathrm{d}\mathbf{z} - \int q(\mathbf{z}) \ln q(\mathbf{z}) \mathrm{d}\mathbf{z} \right\} \\ &\equiv \mathcal{L}(q(\mathbf{z})).\end{aligned} \tag{3}$$

When the free distribution $q(\mathbf{z})$ satisfies the condition $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x})$, the above problem achieves the optimum value, that is the exact value. The variational approximate method restricts the free distribution $q(\mathbf{z})$ to disjoint groups denoted as $\{\mathbf{z}_1, \ldots, \mathbf{z}_s\}$, that is:

$$q(\mathbf{z}) = \prod_{i=1}^{s} q_i(\mathbf{z}_i). \tag{4}$$

The above variational lower bound of the marginal probability $\mathcal{L}(q(\mathbf{z}))$ have the form:

$$\mathcal{L}(q(\mathbf{z})) \propto \int q_j E_{i \neq j}[\ln p(\mathbf{x},\mathbf{z})] \mathrm{d}z_j - \int q_j \ln q_j \mathrm{d}z_j, \tag{5}$$

where

$$E_{i \neq j}[\ln p(\mathbf{x},\mathbf{z})] = \int \ln p(\mathbf{x},\mathbf{z}) \prod_{i \neq j} q_i dz_i. \tag{6}$$

Then maximizing the $\mathcal{L}(q(\mathbf{z}))$ with respect to the distribution $q_j(\mathbf{z}_j)$, we obtain a general expression for the optimization solution $q_j^*(\mathbf{z}_j)$ given by

$$\ln q_j^*(z_j) \propto E_{i \neq j}[\ln p(\mathbf{x},\mathbf{z})]. \tag{7}$$

Through running the variational iteration formula (7) until convergence, we can compute the lower bound of the marginal distribution and the approximate posterior distribution $q^*(\mathbf{z}) = \prod_{j=1}^{s} q_j(\mathbf{z}_j)$ to the distribution $p(\mathbf{z}|\mathbf{x})$.

3 Variational Bayesian extreme learning machine

In this section, we make a detailed description of the variational Bayesian extreme learning machine (VBELM), including the VBELM model, the variational inference procedure, the hidden node selection, and the probabilistic prediction.

3.1 VBELM model

A key concept in the data analysis is that of uncertainty, which arises through noise on measurements and the finite size of data sets. Probability theory provides a consistent framework for the quantification and manipulation of uncertainty. We view the ELM from a probabilistic perspective.

For the sample set $\{\mathbf{x}_i, t_i\}_{i=1}^{N}$, we assume that the target variable t is given by a deterministic function $\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) = \sum_{j=1}^{M} w_j \phi_j(\mathbf{c}_j\mathbf{x} + d_j)$ with additive Gaussian noise that is

$$\begin{aligned}&t = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) + \varepsilon \\ &\varepsilon \sim \mathcal{N}(0, \beta^{-1}).\end{aligned} \tag{8}$$

where ε is a zero-mean Gaussian random variable with variance β^{-1}. This Gaussian distribution is a reasonable hypothesis to express the uncertainty over the data. Then, the variable t satisfies the Gaussian distribution with mean parameter $\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})$ and variance β^{-1}, that is

$$p(t|\mathbf{w}) = \mathcal{N}(t|\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}), \beta^{-1}). \tag{9}$$

According to the independence assumption, the likelihood function of the complete data set $\{\mathbf{x}_i, t_i\}_{i=1}^{N}$ is given by

$$p(\mathbf{t}|\mathbf{w}) = \prod_{i=1}^{N} \mathcal{N}\left(t_i|\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i), \beta^{-1}\right). \tag{10}$$

We can estimate the output weight $\mathbf{w}$ through maximizing the likelihood function, which is equivalent to minimizing the sum-of-squares error function.

To avoid over-fitting problem of maximum likelihood, we adopt the Bayesian perspective, and add a zero-mean Gaussian prior distribution over $\mathbf{w}$ to control the model complexity that is:

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{j=1}^{M} \mathcal{N}\left(w_j|0, \alpha_j^{-1}\right), \tag{11}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_M]^{\mathrm{T}}$, the individual hyperparameter α_i associated independently with each weight w_i, $i = 1, \ldots, M$, respectively. We also add a gamma conjugate prior distribution over the hyperparameter $\boldsymbol{\alpha}$ to attain a full Bayesian model that is

$$p(\boldsymbol{\alpha}) = \prod_{j=1}^{M} \mathrm{Gam}\left(\alpha_j|a_j^0, b_j^0\right), \tag{12}$$

where

$$\mathrm{Gam}(\alpha|a,b) = \Gamma(a)^{-1} b^{a} \alpha^{a-1} \mathrm{e}^{-b\alpha}, \tag{13}$$

with $\Gamma(a) = \int_0^\infty t^{a-1}\mathrm{e}^{-t}\mathrm{d}t$, the 'gamma function.' To make these priors non-informative, we fix their parameters to small values: $a_j^0 = b_j^0 = 10^{-4}$.

Thus, the joint distribution of all the variables $(\mathbf{t}, \mathbf{w}, \boldsymbol{\alpha})$ is given by

$$p(\mathbf{t}, \mathbf{w}, \boldsymbol{\alpha}) = p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})p(\boldsymbol{\alpha}). \tag{14}$$

This is a fully Bayesian treatment of the SLFNs, and the corresponding probabilistic graphical model is shown in Fig. 1. In the model, the circles denote variables, including the output variable t, the output weight vector $\mathbf{w}$, and the hyperparameter vector $\boldsymbol{\alpha}$, and the edges denote the relationship between the nodes. Since the hidden node parameters $\{\mathbf{c}, d\}$ have been assigned randomly, and the input $\mathbf{x}$ are given, then $\phi(\mathbf{cx} + d)$ is fixed. The variance β^{-1} is also fixed through initializing.

The learning problem of the output weights in ELM is transformed to the inference problem of the parameter variable $\mathbf{w}$ in the VBELM model. The inference goal is to compute the posterior distribution of parameter $\mathbf{w}$ and make probabilistic prediction:

- Compute the posterior distribution that is
$$p(\mathbf{w}, \boldsymbol{\alpha}|\mathbf{t}) = \frac{p(\mathbf{t}, \mathbf{w}, \boldsymbol{\alpha})}{p(\mathbf{t})}, p(\mathbf{t}) = \iint p(\mathbf{t}, \mathbf{w}, \boldsymbol{\alpha})\mathrm{d}\mathbf{w}\mathrm{d}\boldsymbol{\alpha}. \tag{15}$$
- Make probabilistic prediction under new input $\mathbf{x}^*$ given the sample $\{\mathbf{x}_i, t_i\}_{i=1}^{N}$ that is
$$p(t|\mathbf{x}^*, \mathbf{t}) = \int p(t|\mathbf{w}, \mathbf{x}^*)p(\mathbf{w}|\mathbf{t})\mathrm{d}\mathbf{w}. \tag{16}$$

Above marginalization over the parameters $\mathbf{w}, \boldsymbol{\alpha}$ in computing $p(t)$ and $p(\mathbf{w}|\mathbf{x})$ is analytically intractable. Thus, we adapt the variational method to approximate the posterior distribution and make probabilistic prediction.

3.2 Variational inference

The variational framework provides a flexible way to approximate the posterior distribution. We introduce the factorized free distribution $q(\mathbf{w}, \boldsymbol{\alpha}) = q(\mathbf{w})q(\boldsymbol{\alpha})$ to approximate the posterior distribution $p(\mathbf{w}, \boldsymbol{\alpha}|\mathbf{t})$. According to the variational iteration formulas (7), we obtain the iteration formulas of the approximate distributions, respectively:

$$\ln q^*(\boldsymbol{\alpha}) \propto E_{q(\mathbf{w})}[\ln p(\mathbf{t}, \mathbf{w}, \boldsymbol{\alpha})], \tag{17}$$

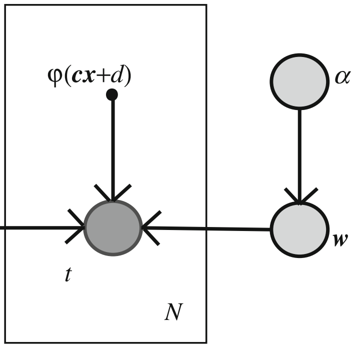

Fig. 1 Probabilistic graphical model of the fully Bayesian SLFN

$$\ln q^*(\mathbf{w}) \propto E_{q(\boldsymbol{\alpha})}[\ln p(\mathbf{t}, \mathbf{w}, \boldsymbol{\alpha})]. \tag{18}$$

Thus, we can obtain the updates of the posterior distributions. The iteration formulas of $q(\alpha_j)$, $j = 1, \ldots, M$ are Gamma distributions:

$$\begin{aligned} q^*(\alpha_j) &= \mathrm{Gam}\left(\alpha_j|a_j, b_j\right) \\ a_j &= a_j^0 + \frac{1}{2} \\ b_j &= b_j^0 + \frac{1}{2}E_{q(\mathbf{w})}\left[w_j^2\right], \end{aligned} \tag{19}$$

where $E_{q(\mathbf{w})}\left[w_j^2\right] = \mathbf{m}(j)\mathbf{m}(j) + \mathbf{S}(j,j)$. The iteration formulas of the $q(\mathbf{w})$ is a M-dimensional Gaussian distribution:

$$\begin{aligned} q^*(\mathbf{w}) &= \mathcal{N}(\mathbf{w}|\mathbf{m}, \mathbf{S}) \\ \mathbf{m} &= \beta\mathbf{S}\Phi^{\mathrm{T}}\mathbf{t} \\ \mathbf{S} &= (\mathbf{A} + \beta\boldsymbol{\Phi}^{T}\boldsymbol{\Phi})^{-1}, \end{aligned} \tag{20}$$

where $\mathbf{A} = \mathrm{diag}(a_1/b_1, \ldots, a_M/b_M)$. The variational approximate inference is processed by updating the approximate posterior distribution $q(\boldsymbol{\alpha}), q(\mathbf{w})$ with the Eqs. (19) and (20) alternately until the convergence criteria have been satisfied.

The variational inference of VBELM has two calculation steps: First, choose the input weight matrix $[\mathbf{c}_1, \ldots, \mathbf{c}_M]$ and threshold vector $[d_1, \ldots, d_M]$ randomly and then compute the posterior distribution of output weight $\mathbf{w}$ using the above variational iteration formulae (19) and (20). The hyperparameter variance β^{-1} denotes the uncertainty of the samples, which should be initialized with a small value, like $\beta^{-1} = 0.1$. The hyperparameter a_0, b_0 are the parameters of the prior parameter $\boldsymbol{\alpha}$. We fix their parameters to small values: $a_0 = b_0 = 10^{-4}$ to make these priors non-informative. The inference procedure of VBELM is shown in the Algorithm 1.

Algorithm 1: The variational inference procedure of VBELM

```
Data: Dataset $\{\mathbf{x}_i, t_i\}_{i=1}^{N}$, number of hidden node $M$, activation function $f(\ )$
Result: $q(\mathbf{w}) = N\ (\mathbf{w} | \mathbf{m}, \mathbf{S})$.
begin
    % Choose the input weight and threshold randomly
    Input=rand($n$,$M$);
    Thre=rand(1,$M$);
    $\mathbf{\Phi}'_{N\times M} = (\mathbf{x}_1, \cdots, \mathbf{x}_N)^{\mathrm{T}} * \mathrm{Input+Thre}$;
    $\mathbf{\Phi}_{N\times M} = f(\mathbf{\Phi}')$.
    % Train the output weight with variational method
    Initialization the hyperparameters $a^0 = b^0 = 10^{-4}, \beta^{-1} = 0.1$;
    for $i \leftarrow 1$ to $|a^i - a^{i-1}| < 10^{-4}$ do
        $\mathbf{S} = (\mathbf{A} + \beta \mathbf{\Phi}^T \mathbf{\Phi})^{-1}$;
        $\mathbf{m} = \beta \mathbf{S} \mathbf{\Phi}^{\mathrm{T}} \mathbf{t}$;
        for $j \leftarrow 1$ to $M$ do
            $a_j = a_j^0 + \frac{1}{2}$;
            $b_j = b_j^0 + \frac{1}{2} * (\mathbf{m}^2(j) + \mathbf{S}(j, j))$;
        end
        $\mathbf{A} = \mathrm{diag}(a_1 / b_1, \cdots, a_M / b_M)$;
    end
    Output $q(\mathbf{w}) = N\ (\mathbf{w} | \mathbf{m}, \mathbf{S})$.
end
```

3.3 Hidden node selection

In the VBELM model, the output weight $\mathbf{w} = [w_1, \ldots, w_M]^{\mathrm{T}}$ has a zero-mean Gaussian conjugate prior distribution $p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{j=1}^{M} \mathcal{N}\left(w_j | 0, \alpha_j^{-1}\right)$. The hyperparameter $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_M]$ has a Gamma prior distribution $p(\alpha_j) = \mathrm{Gam}\left(\alpha_j | a_j^0, b_j^0\right)$ with parameters $\left\{a_j^0, b_j^0\right\}_{j=1}^{M}$. Since the prior distributions are conjugate priors, the posterior distribution $q(\mathbf{w})$ satisfies Gaussian distribution, and the posterior distribution $q(\boldsymbol{\alpha})$ satisfies Gamma distribution, which have the distributions:

$$q(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}, \mathbf{S}), \tag{21}$$

$$q(\alpha_j) = \mathrm{Gam}(\alpha_j | a_j, b_j) \quad \forall j. \tag{22}$$

The hyperparameters $\{\alpha_1, \ldots, \alpha_M\}$ are associated with the output weights $\{w_1, \ldots, w_M\}$, respectively, and they determine the importance of hidden nodes in the VBELM model. If the value a_i/b_i tends to infinity, the corresponding ith element in the matrix $\mathbf{S} = (\mathbf{A} + \beta \mathbf{\Phi}^T \mathbf{\Phi})^{-1}$ tends to zero, $\mathbf{S}(i, i) \to 0$, and the mean value tends to zero $m(i) \to 0$. The posterior approximate distribution $q(w_i|\mathbf{t}, \boldsymbol{\alpha})$ becomes highly peaked at zero. Thus, the corresponding hidden nodes can be 'pruned' from the model. The procedure of pruning the hidden nodes is shown in the Algorithm 2.

Algorithm 2: Hidden node selection of VBELM

```
Data: Posterior distribution $q(\mathbf{w}) = N\ (\mathbf{w} | \mathbf{m}, \mathbf{S})$
Result: Pruned posterior distribution $q(\mathbf{w}) = N\ (\mathbf{w} | \mathbf{m}_P, \mathbf{S}_P)$
begin
    for $j \leftarrow 1$ to $M$ do
        if $a_i / b_i >$ best criterion then
            Prune the hidden node $w_i$;
            Delete the corresponding rows and columns in $\mathbf{m}, \mathbf{S}$;
        end
        Obtain the pruned parameters: $\mathbf{m}_P, \mathbf{S}_P$;
    end
    Output $q(\mathbf{w}) = N\ (\mathbf{w} | \mathbf{m}_P, \mathbf{S}_P)$.
end
```

3.4 Probabilistic prediction

We can make predictions based on the approximate posterior distribution $q(\mathbf{w})$. Given a new input $\mathbf{x}^*$, the predictive distribution of $\boldsymbol{t}$ can be approximate with the distribution $p(t|\mathbf{w})$ in the Eq. (8) and the approximate posterior distribution $q(\mathbf{w})$ in the Eq. (20) that is

$$\begin{aligned} p(t|x^*, \mathbf{t}) &= \int p(t|\mathbf{x}^*, \mathbf{w}) p(\mathbf{w}|\mathbf{t}) \mathrm{d}\mathbf{w} \\ &\approx \int p(t|\mathbf{x}^*, \mathbf{w}) q(\mathbf{w}) \mathrm{d}\mathbf{w}. \end{aligned} \tag{23}$$

Since both terms in the integrand are Gaussian, the predictive distribution is also a Gaussian distribution, giving:

$$p(t|\mathbf{x}^*, \mathbf{t}) = \mathcal{N}(t|\mu(\mathbf{x}^*), \delta^2(\mathbf{x}^*)), \tag{24}$$

with

$$\mu(\mathbf{x}^*) = \mathbf{m}^{\mathrm{T}} \phi(\mathbf{x}^*), \tag{25}$$

$$\delta^2(\mathbf{x}^*) = \frac{1}{\beta} + \phi(\mathbf{x}^*)^{\mathrm{T}} \mathbf{S} \phi(\mathbf{x}^*). \tag{26}$$

Thus, the predictive distribution of $\mathbf{x}^*$ has a mean $\mathbf{m}^{\mathrm{T}} \phi(\mathbf{x}^*)$ with a variance $\frac{1}{\beta} + \phi(\mathbf{x}^*)^{\mathrm{T}} \mathbf{S} \phi(\mathbf{x}^*)$. Comparison with Eq. (9), we can see that the predictive output is weighted by the posterior mean weights $\mathbf{m}$. The predictive variance contains two terms: the noise on the data $\frac{1}{\beta}$ and the uncertainty in the weight prediction $\phi(\mathbf{x}^*)^{\mathrm{T}}\mathbf{S}\phi(\mathbf{x}^*)$. The prediction of the VBELM algorithm is shown in Algorithm 3.

Algorithm 3: Probabilistic prediction of VBELM

```
Data: Posterior distribution $q(\mathbf{w}) = N(\mathbf{w} \mid \mathbf{m}, \mathbf{S})$, new input $\mathbf{x}^*$
Result: Prediction $p(t \mid \mathbf{x}^*, \mathbf{t})$
begin
    Compute the prediction mean $\mu$ with the equation (25);
    Compute the prediction variance $\delta^2$ with the equation (26);
    Output the prediction distribution $p(t \mid \mathbf{x}^*, \mathbf{t})$.
end
```

4 Experiments

In this section, we present some experiments of the VBELM applied to data sets in both regression and classification.

4.1 VBELM regression: 'SinC' function

The 'SinC' function has the form of

$$y(x) = \begin{cases} \dfrac{\sin(x)}{x}, & x \neq 0 \\ 1, & x = 0 \end{cases}. \tag{27}$$

The function has been a popular choice to variable regression. We make the VBELM regression in the 'SinC' synthetic data sets and compare it with the traditional ELM regression.

4.1.1 Experiment 1: Regression

We generate 2,000 sample set $\{x_i, y_i\}$ using the 'SinC' function, where x_i's are uniformly randomly distribution on the interval $(-10, 10)$, and y_i's are added in a zero-mean Gaussian noise with variance 0.2. For the VBELM model, we set $M = 100$ hidden node parameters with $c_i \in [-10, 10]$, $d_i \in [0, 10]$ randomly, assign the parameters $\beta^{-1} = 0.1$ and $a^0 = b^0 = 10^{-4}$, and select the sigmoid function as the activation function. We make VBELM regression in the data sets. The results are shown in the Fig. 2, where the red solid line denotes the exact 'SinC' function, the *black dotted line* is the mean of the Gaussian predictive distribution of the VBELM algorithm, the gray shaded region spans one standard deviation either side of the mean.

The experiments show that the VBELM algorithm presents a Gaussian predictive distribution compared to the tradition point estimation, where the mean values provide the predictive values with highest probability, while the variances present the uncertainty of the prediction came from the data noise and the output weight. Concretely, the Fig. 2 shows that the means of the predictive distribution of VBELM has a high approximate accuracy, the region spanned one standard deviation either side of the mean covers the more than 95 % data.

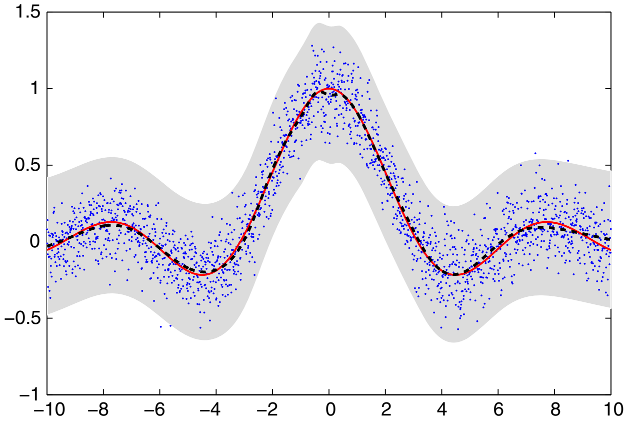

Fig. 2 The VBELM regression in the 'SinC' synthetic data sets. The *red solid line* denotes the exact 'SinC' function, the *black dotted line* is the mean of the Gaussian predictive distribution of the VBELM algorithm, and the *gray shaded region* spans one standard deviation either side of the mean (color figure online)

4.1.2 Experiment 2: Comparison

We generate 5,000 'SinC' testing data set $\{x_i, y_i\}$, where x_i's are uniformly randomly distributed on the interval $(-10, 10)$ and y_i's are generated from the 'SinC' function. Besides, we generate 5,000 training data set $\{x_i, y_i\}$, where x_i's are uniformly randomly distributed on the interval $(-10, 10)$ and y_i's are added in zero-mean Gaussian noise with variance 0.2.

We compare the ELM and VBELM regression under different random hidden node parameters. We set $M = 20$ hidden nodes with four type of hidden node parameters $c_i \in [-1, 1], d_i \in [0, 1], c_i \in [-10, 10], d_i \in [0, 10], c_i \in [-20, 20], d_i \in [0, 20]$, and $c_i \in [-30, 30], d_i \in [0, 30]$, respectively, and select the sigmoid function as the activation function. Then, we execute 100 trials and compute the average value of the error (root mean square error), the running time, and the 2-norm of the output weights. The comparison results are shown in Table 1 and Fig. 3. The experiment results show that: (1) The testing errors of ELM increase 20 times when the random parameters change from $c_i \in [-1, 1], d_i \in [0, 1]$ to $c_i \in [-30, 30], d_i \in [0, 30]$, while the testing errors of VBELM increase <10 times. (2) The 2-norm of output weights $\|\boldsymbol{w}\|_2$ of ELM are 10^3–10^4 larger than the values of VBELM in average.

Table 1 Performance comparison of ELM and VBELM for 'SinC' function regression

Algorithms	Error		Time (s)		$\|\|\boldsymbol{w}\|\|_2$	Hidden node parameters ($M = 20$)
	Training	Testing	Training	Testing		
ELM	0.1417	0.0075	0.0359	0.0153	8.85×10^6	$\boldsymbol{c}_i \in [-1, 1]$, $d_i \in [0, 1]$
VBELM	0.1424	0.0123	0.3825	0.0144	230	
ELM	0.1616	0.0766	0.0363	0.0125	0.54×10^3	$\boldsymbol{c}_i \in [-10, 10]$, $d_i \in [0, 10]$
VBELM	0.1620	0.0767	0.3888	0.0134	110	
ELM	0.1762	0.1035	0.0359	0.0144	4.83×10^3	$\boldsymbol{c}_i \in [-20, 20]$, $d_i \in [0, 20]$
VBELM	0.1725	0.0973	0.3866	0.0141	21	
ELM	0.1869	0.1211	0.0369	0.0134	4.21×10^4	$\boldsymbol{c}_i \in [-30, 30]$, $d_i \in [0, 30]$
VBELM	0.1838	0.1159	0.3941	0.0075	34	

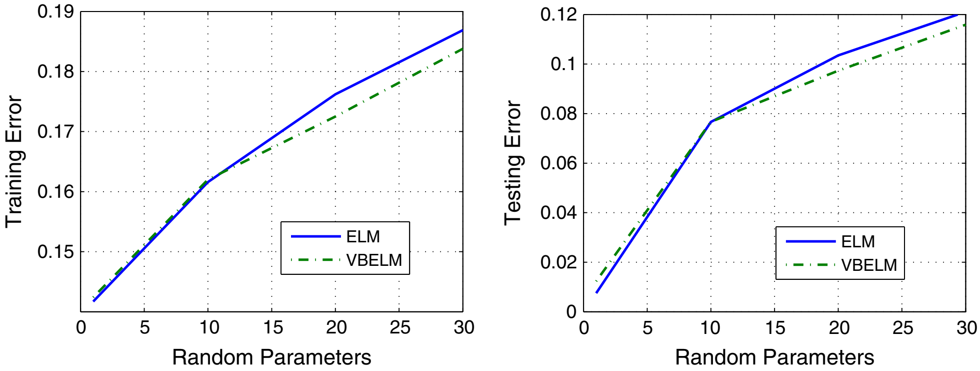

Fig. 3 Performance comparison of ELM and VBELM for 'SinC' function regression

Error comparison: ELM and VBELM have similar training error and testing error for smaller random parameters of hidden nodes, while the VBELM has lower error than the ELM algorithm for the larger random parameters. This shows that the VBELM algorithm has higher stability than ELM for different random assignment of hidden node parameters. *2-Norm comparison:* The 2-norm of output weights $\boldsymbol{w}$ of ELM is very large compared with VBELM, which illustrates that the generalization inverse of hidden node parameter matrix $\boldsymbol{\Phi}$ is ill-posed, and this would cause weak stabler and week generalization performance. *Time comparison:* Both of the algorithms have the same running time in testing, and ELM runs 10 times faster than VBELM in training, which consumes more time in the iteration with prior distribution.

We can conclude that the VBELM is stabler for different assignment of hidden parameters and has better generalization performance than ELM, but has slower training speed. This shows that the VBELM algorithm controls the over-fitting and the unstability of ELM through involving the variational framework with prior distribution, and this also consumes more time in the variational iteration process.

4.2 VBELM classification

In this section, we compare the VBELM, ELM, and SVM algorithms in three types of standard data sets from the UCI machine learning repository—Iris, Lance-sca, and Pima Indians Diabetes [23]. The attributions of the standard data sets are described in the Table 2, where the concrete quality 60 (20 + 20 + 20) denotes that the total number of training samples is 60 and each class has 20 samples, respectively.

The VBELM and ELM algorithms are random methods, so the results are different with each random assignment of hidden node parameters. We execute 100 trails and compute the average times and average accuracies for each groups of data in the experiments. In this section, we compare the accuracy and the stability of algorithms under different numbers of hidden nodes, where the hidden node numbers are assigned manually and fixed at the beginning of the experiments.

4.2.1 Comparison of ELM and VBELM

We compare the VBELM and ELM algorithm in the standard data sets under different hidden number $M = 20$,

Table 2 The attributes description of the standard data sets

Data set	Training sample	Testing sample	Attribute number	Class number	Data characteristics
Iris	60 (20 + 20 + 20)	90 (30 + 30 + 30)	4	3	Multivariate
Lance-sca	319 (25 + 151 + 143)	306 (24 + 137 + 145)	13	3	Multivariate
Pima Indians diabetes	384 (250 + 134)	384 (250 + 134)	8	2	Binary

Table 3 Classification comparison on the standard sets with hidden node number $M = 20$

Data sets	Algorithms	Time (s)		Accuracy		Hidden node parameters ($M = 20$)
		Training	Testing	Training	Testing	
Iris	ELM	0.0023	0.0031	0.9997	0.9478	$\boldsymbol{c}_i \in [-1, 1]$, $d_i \in [0, 1]$
	VBELM	0.0169	0.0017	0.9267	0.9069	
	ELM	0.0023	0.0027	0.8873	0.7991	$\boldsymbol{c}_i \in [-10, 10]$, $d_i \in [0, 10]$
	VBELM	0.0173	0.0023	0.8525	0.8000	
Balance-sca	ELM	0.0042	0.0051	0.9125	0.7220	$\boldsymbol{c}_i \in [-1, 1]$, $d_i \in [0, 1]$
	VBELM	0.0315	0.0045	0.8835	0.7390	
	ELM	0.0055	0.0028	0.8387	0.7325	$\boldsymbol{c}_i \in [-10, 10]$, $d_i \in [0, 10]$
	VBELM	0.0320	0.0022	0.8142	0.7315	
Pima Indians diabetes	ELM	0.0069	0.0053	0.6770	0.6445	$\boldsymbol{c}_i \in [-1, 1]$, $d_i \in [0, 1]$
	VBELM	0.0348	0.0023	0.8829	0.7569	
	ELM	0.0064	0.0034	0.6707	0.6421	$\boldsymbol{c}_i \in [-10, 10]$, $d_i \in [0, 10]$
	VBELM	0.0300	0.0020	0.8143	0.7199	

$M = 50$, $M = 100$ and different hidden node parameters $c_i \in [-1, 1], d_i \in [0, 1]$ and $c_i \in [-10, 10], d_i \in [0, 10]$, which generate 18 groups of comparison data sets. Since random assignments of the weights $\{\mathbf{c}_i, d_i\}$ make the learning results varied for VBELM and ELM, we execute 100 trials and compute the average running times and average accuracies for each group of data. The comparison results are shown in the Tables 3, 4, 5 and Fig. 4.

The Tables 3, 4, and 5 show that for the total 18 groups of training sets, the VBELM algorithm provides higher accuracy in 6 groups than ELM; while for the total 18 groups of testing sets, the VBELM algorithm provides higher accuracy in 17 groups than ELM. The ELM is preferable in training sets most of the time, but is worse for almost all of the testing sets than VBELM algorithm. Concretely, for Iris and Balance-sca data sets, the ELM algorithm is preferable to the VBELM in the training sets, but is worse in the corresponding testing sets. For the Pima Indians Diabetes data sets, the VBELM algorithm has higher accuracy both in the training and the testing sets. All this show that the VBELM algorithm has better generalization performance than ELM.

The Fig. 4 describes the accuracy changes of VBELM and ELM algorithm with the number M and the hidden node parameters $\{\boldsymbol{c}, d\}$ varying. The Fig. 4 shows that (1) The training accuracy of ELM becomes higher with the number M increasing, but the testing accuracy of ELM becomes worse for all of the testing sets with M increasing, which show that the ELM has the over-fitting problem. (2) The training accuracy and the testing accuracy of ELM become unstable when the hidden node parameters $\{\boldsymbol{c}, d\}$ become lager, especially the testing accuracy declines to near 60 %, which is poor for prediction. (3) The training accuracy and testing accuracy of VBELM are improved with the M increasing, which show that the VBELM algorithm has strong generalization capability. (4) The training accuracy and testing accuracy of VBELM are stable when the hidden node parameters $\{\boldsymbol{c}, d\}$ change, which demonstrate that the VBELM has stable performance.

4.2.2 *Comparison of VBELM, ELM, and SVM*

We also compare the VBELM, ELM and SVM algorithm in the standard data sets, where the parameters of VBELM and ELM are $M = 100$ and $c_i \in [-1, 1], d_i \in [0, 1]$. The comparison results shown in the Table 6 illustrate that (1) VBELM has the highest testing accuracy for all the three data sets. (2) SVM has better performance in training accuracy for the Iris and Pima Indians Diabetes data sets, while ELM has better performance in training accuracy for the Iris and Balance-sca data sets. (3) For the training accuracy performance, we have SVM $\approx$ ELM $>$ VBELM, while for the testing accuracy performance, we have

Table 4 Classification comparison on the standard sets with hidden node number $M = 50$

Data sets	Algorithms	Time (s)		Accuracy		Hidden node parameters ($M = 50$)
		Training	Testing	Training	Testing	
Iris	ELM	0.0033	0.0020	1.0000	0.8948	$\boldsymbol{c}_i \in [-1, 1]$, $d_i \in [0, 1]$
	VBELM	0.0464	0.0052	0.9853	0.9494	
	ELM	0.0031	0.0048	0.9785	0.8294	$\boldsymbol{c}_i \in [-10, 10]$, $d_i \in [0, 10]$
	VBELM	0.0442	0.0030	0.9352	0.8883	
Balance-sca	ELM	0.0098	0.0041	0.9308	0.6983	$\boldsymbol{c}_i \in [-1, 1]$, $d_i \in [0, 1]$
	VBELM	0.1006	0.0034	0.9045	0.7417	
	ELM	0.0111	0.0031	0.9045	0.7549	$\boldsymbol{c}_i \in [-10, 10]$, $d_i \in [0, 10]$
	VBELM	0.1050	0.0055	0.8691	0.7611	
Pima Indians diabetes	ELM	0.0127	0.0033	0.7113	0.6505	$\boldsymbol{c}_i \in [-1, 1]$, $d_i \in [0, 1]$
	VBELM	0.0825	0.0019	0.9052	0.7404	
	ELM	0.0164	0.0042	0.7102	0.6468	$\boldsymbol{c}_i \in [-10, 10]$, $d_i \in [0, 10]$
	VBELM	0.0856	0.0041	0.8692	0.7651	

Table 5 Classification comparison on the standard sets with hidden node number $M = 100$

Data sets	Algorithms	Time (s)		Accuracy		Hidden node parameters ($M = 100$)
		Training	Testing	Training	Testing	
Iris	ELM	0.0083	0.0027	1.0000	0.8970	$\boldsymbol{c}_i \in [-1, 1]$, $d_i \in [0, 1]$
	VBELM	0.1269	0.0023	0.9980	0.9541	
	ELM	0.0083	0.0045	0.9997	0.5731	$\boldsymbol{c}_i \in [-10, 10]$, $d_i \in [0, 10]$
	VBELM	0.1156	0.0078	0.9692	0.9177	
Balance-sca	ELM	0.0320	0.0122	0.9613	0.5243	$\boldsymbol{c}_i \in [-1, 1]$, $d_i \in [0, 1]$
	VBELM	0.3170	0.0067	0.9106	0.7577	
	ELM	0.0328	0.0100	0.9536	0.7609	$\boldsymbol{c}_i \in [-10, 10]$, $d_i \in [0, 10]$
	VBELM	0.3115	0.0056	0.9061	0.8029	
Pima Indians diabetes	ELM	0.0272	0.0052	0.7573	0.6371	$\boldsymbol{c}_i \in [-1, 1]$, $d_i \in [0, 1]$
	VBELM	0.2322	0.0063	0.9101	0.7657	
	ELM	0.0323	0.0023	0.7441	0.6371	$\boldsymbol{c}_i \in [-10, 10]$, $d_i \in [0, 10]$
	VBELM	0.2236	0.0031	0.9087	0.7915	

VBELM > SVM > ELM. We can conclude that the VBELM algorithm has better generalization than ELM and SVM for the classification problem in the standard data sets.

4.3 Hidden nodes selection

In Sect. 3.3, we have shown that the values a_i/b_i can be used to select the hidden nodes i. If the values $a_i/b_i >$ best criterion, the corresponding hidden nodes can be 'pruned' from the model. For the three standard data sets, we assign the initialization value of the hidden node number with $M = 200$, then we use different criteria to prune the hidden node, respectively, and then we analyze the classification accuracy with different hidden node numbers.

For the three standard data sets—Iris, Balance-sca, Pima Indians Diabetes, we first compute the posterior distribution $q(\mathbf{w})$ and $q(\boldsymbol{\alpha})$ according to the Eqs. (19) and (20), where $\mathbf{A} = \mathrm{diag}(a_1/b_1, \ldots, a_M/b_M)$. Then, we prune the hidden nodes with different criteria, and the experiment results are shown in the Table 7 and Fig. 5. Accuracy comparisons under different hidden node numbers. The x axis denotes the number of left hidden nodes, the y axis denotes the classification accuracy, the solid lines denote training accuracy, the dotted lines denote testing accuracy. The experiments show that (1) For the Iria data set, the maximum $\max_c = \max\{a_i/b_i | i = 1, \ldots, M\}$ is 7.06, the minimum $\mathrm{mic}_c = \min\{a_i/b_i | i = 1, \ldots, M\}$ is 0.06, and the ratio $\frac{\max_c}{\mathrm{mic}_c}$ has the order of 10^2, which indicates the range of the set $\{a_i/b_i\}_{i=1}^{M}$. The classification accuracies (including training accuracies and testing accuracies) decrease rapidly with the hidden node pruned, and there is no the best hidden node number. (2) For the Balance-sca

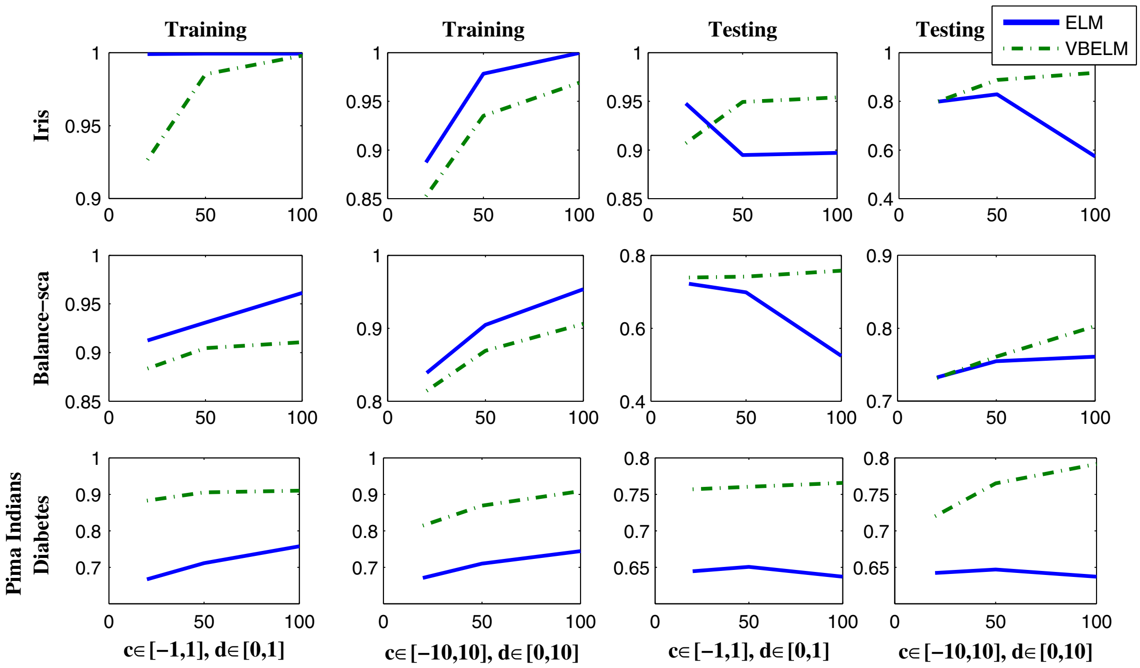

Fig. 4 Performance comparison of ELM and VBELM in the classification of the standard data sets, where the *solid lines* denote the ELM algorithm, and the *dash dot lines* denote the VBELM algorithm, the *x* axis indicates the number *M* of hidden nodes, the *y* axis indicates the accuracy of algorithms

Table 6 Comparison of classification accuracy on the standard sets

Algorithms	Iris		Balance-sca		Pima Indians diabetes	
	Accuracy		Accuracy		Accuracy	
	Training	Testing	Training	Testing	Training	Testing
VBELM	0.9980	*0.9541*	0.9106	*0.7577*	0.9101	*0.7657*
ELM	1.0000	0.8970	0.9613	0.5243	0.7573	0.6371
SVM	1.0000	0.9444	0.9341	0.6797	1.0000	0.6510

The values in italics indicate the testing accuracies of VBELM algorithm

data set, we have $\max_c = 107.48$, $\text{mic}_c = 0.03$ and the ratio has the order of 10^4. There is a turning point at the value $c_i > 38, M_P = 30$, which shows that $c_i > 38$ is the best criterion, and the corresponding 30 hidden nodes can be pruned without loss of accuracy. (3) For the Pima Indians Diabetes data set, we have $\max_c = 704.20$, $\text{mic}_c = 0.03$, and the ratio has a order of 10^5. There is also a turning point at the value $c_i > 160, M_P = 49$. (4) When the parameter set $\{a_i/b_i\}_{i=1}^{M}$ has wide range, like the order $\frac{\max_c}{\text{mic}_c}$ tends to 10^4, 10^5 in the Balance-sca and Pima Indians Diabetes data sets, we can find the turning point using the criteria a_i/b_i, that is, the optimum hidden node number. Moreover, the wider range the set $\{a_i/b_i\}_{i=1}^{M}$ has, the better performance the criterion provides in the hidden node selection. (5) In this experiment, we set the initialization of the hidden node number with $M = 200$ and get the accuracies under the pruned hidden nodes. Compared with the initial hidden node number fixed in Sect. 4.2, we can see that the VBELM under pruned hidden nodes provides almost the same accuracies with the fixed hidden nodes $M = 100$ in Sect. 4.2. Hidden node selection based on the VBELM is also related with the initialization of the hidden node number, and different initializations lead to different best number of hidden nodes, which is one of the interesting directions for future research.

We can conclude that the value a_i/b_i in the hyperparameter distribution $q^*(\alpha_i) = \text{Gam}(\alpha_i|a_i, b_i)$ can be used to select the hidden nodes as criterion. Moreover, for the data sets with wide range in the set $\{a_i/b_i|i = 1, \ldots, M\}$, the hidden node selection with the criterion has better performance.

4.4 Experiments conclusion

The regression experiments 4.1 show that the VBELM presents a Gaussian predictive distribution compared to the traditional point estimation, which has enriched the

Table 7 Accuracy comparisons under different hidden node numbers

Iris				Balance-sca				Pima Indians diabetes			
$\max_c = 7.06$, $\min_c = 0.06$, $\frac{\max_c}{\text{mic}_c} \to 10^2$				$\max_c = 107.48$, $\min_c = 0.03$, $\frac{\max_c}{\text{mic}_c} \to 10^4$				$\max_c = 704.20$, $\min_c = 0.03$, $\frac{\max_c}{\text{mic}_c} \to 10^5$			
Criteria	N_p	Train-A	Test-A	Criteria	N_p	Train-A	Test-A	Criteria	N_p	Train-A	Test-A
No prune	0	0.9999	0.9586	No prune	0	0.9172	0.7821	No prune	0	0.8001	0.6408
$c_i > 4.7$	10	0.9025	0.8621	$c_i > 53$	10	0.9165	0.7635	$c_i > 480$	10	0.8000	0.6409
$c_i > 3.7$	20	0.7662	0.7419	$c_i > 42$	20	0.9113	0.7444	$c_i > 360$	20	0.8000	0.6408
$c_i > 3.2$	31	0.5428	0.5291	$c_i > 38$	*30*	*0.9008*	*0.7240*	$c_i > 300$	30	0.7995	0.6407
$c_i > 2.0$	52	0.3732	0.3715	$c_i > 30$	50	0.8814	0.6869	$c_i > 160$	*49*	*0.7976*	*0.6396*
$c_i > 0.85$	80	0.3289	0.3310	$c_i > 18$	83	0.8394	0.6545	$c_i > 33$	80	0.7640	0.6236
$c_i > 0.45$	100	0.3329	0.3336	$c_i > 12$	104	0.8032	0.6427	$c_i > 13$	100	0.6738	0.5782

The original hidden node number $M = 200$, criteria $c_i = a_i/b_i$, $\max_c = \max\{c_i|i = 1, \ldots, M\}$, $\min_c = \min\{c_i|i = 1, \ldots, M\}$, N_p denotes the number of pruned hidden nodes, Train-A denotes the training accuracy, Test-A denotes the testing accuracy

The values in italics indicate the turning points of the hidden node numbers

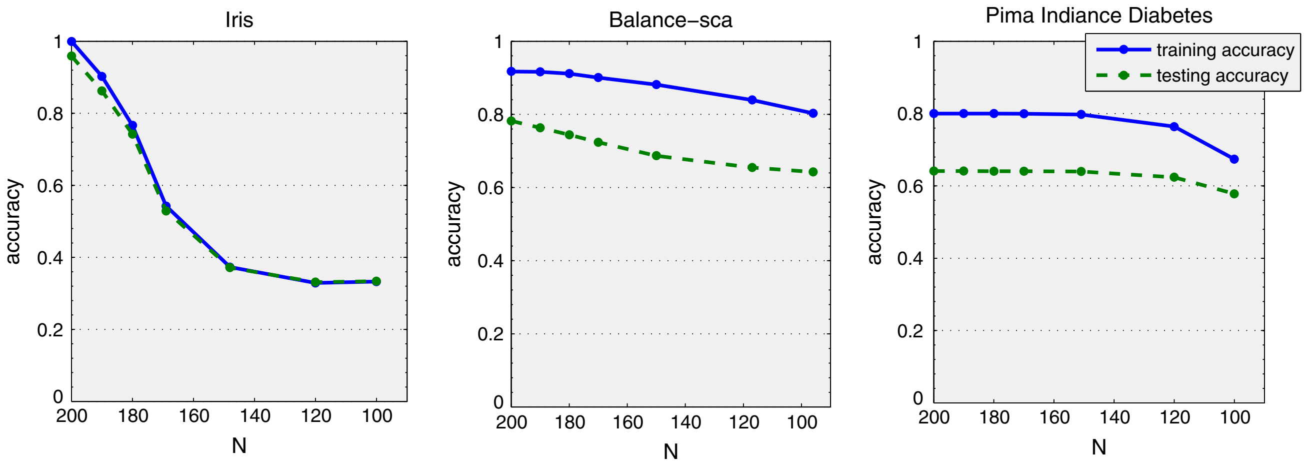

Fig. 5 Accuracy comparisons under different hidden node numbers. The *x* axis denotes the number of left hidden nodes, the *y* axis denotes the classification accuracy, the *solid lines* denote training accuracy, the *dotted lines* denote testing accuracy

predictive information. Moreover, the results also show that VBELM algorithm is stable for different assignment of hidden node parameters and has better generalization performance in 'SinC' function regression compared with ELM algorithm.

The classification experiments 4.2 show that the VBELM algorithm has better generalization than ELM and SVM for classification in all the three standard sets. Besides, the VBELM algorithm is stable than ELM for different assignments of the hidden node parameters $\{\mathbf{c}, \mathbf{d}\}$. All This shows that the VBELM algorithm improves the generalization and stability through the variational Bayesian framework.

The hidden node selection experiments 4.3 show that the value a_i/b_i in the hyperparameter distribution $q^*(\alpha_i) = \text{Gam}(\alpha_i|a_i, b_i)$ is a effective criterion for hidden node selection in VBELM. Especially, for the data sets with wide range in the set $\{a_i/b_i|i = 1, \ldots, M\}$, the hidden node selection with the criterion has better performance.

5 Conclusions

In this paper, we have designed the variational Bayesian extreme learning machine to solve the ill-posed problem and the automatical selection problem of the hidden nodes. First, we add the model complexity information with Bayesian prior distribution over output weights, which tends to control the ill-posed problem and over-fitting problem of ELM. Then, we employ the variational approximation for the Bayesian framework to compute the posterior distribution of output weights and the variational hyperparameters, where the independent variational hyperparameters can be applied to hidden node selection. Theoretical analysis and experimental results elucidate that the variational framework provides stabler error and better generalization performance, simultaneously the variational hyperparameters can be used to select the hidden nodes.

There are several interesting directions for future research. Firstly, the selection of hidden nodes is an important research direction, which describes the model complexity of VBELM/ELM. In this paper, we adapt the hyperparameters as criteria in variational inference procedure to prune the hidden nodes, but how to select the optimal criteria automatically is a further research direction. Secondly, ELM is a supervised learning method with extremely fast learning speed. It is promising to bring together the ELM and the unsupervised learning methods, using the variational inference especially. Thus, the speed advantage of ELM can be used to unsupervised learning procedure. Thirdly, Markov random fields can model the images directly using pixels, which contain full information than the preprocessing method like PCA and LBP. We attempt to combine the ELM and Markov random fields in the image recognition application.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China under Grant 61063035, 61402332, and the young academic team construction projects of the 'twelve five' integrated investment planning in Tianjin University of Science and Technology.

References

1. Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of international joint conference on neural networks (IJCNN2004), vol 2, (Budapest, Hungary), pp 985–990
2. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
3. Zhang R, Huang GB, Sundararajan N, Saratchandran P (2007) Multi-category classification using an extreme learning machine for microarray gene expression cancer diagnosis. IEEE/ACM Trans Comput Biol Bioinf 4(3):485–495
4. Mattos CLC, Barreto GA (2013) ARTIE and MUSCLE models: building ensemble classifiers from fuzzy ART and SOM networks. Neural Comput Appl 22(1):49–61
5. Huang GB, Chen L, Siew CK (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17(4):879–892
6. Han F, Huang DS (2006) Improved extreme learning machine for function approximation by encoding a priori information. Neurocomputing 69(16):2369–2373
7. Tang X, Han M (2009) Partial Lanczos extreme learning machine for single-output regression problems. Neurocomputing 72(13): 3066–3076
8. Minhas R, Baradarani A, Seifzadeh S, Jonathan Wu QM (2010) Human action recognition using extreme learning machine based on visual vocabularies. Neurocomputing 73(10):1906–1917
9. Malathi V, Marimuthu NS, Baskar S (2010) Intelligent approaches using support vector machine and extreme learning machine for transmission line protection. Neurocomputing 73(10):2160–2167
10. Yeu CW, Lim MH, Huang GB, Agarwal A, Ong YS (2006) A new machine learning paradigm for terrain reconstruction. Geosci Remote Sens Lett IEEE 3(3):382–386
11. Huang GB, Chen L (2008) Enhanced random search based incremental extreme learning machine. Neurocomputing 71: 3460–3468
12. Saul LK, Jaakkola TS, Jordan MI (1996) Mean field theory for Sigmoid belief networks. J Artif Intell Res 4:61–76
13. Yedidia JS, Freeman WT, Weiss Y (2001) Bethe free energy, Kikuchi approximations and belief propagation algorithms. Technical Report, TR-2001-16, Mitsubishi Electric Research Laboratories, Cambridge
14. Yedidia JS, Freeman WT, Weiss Y (2001) Generalized belief propagation. In: Advances in neural information processing systems 13, vol 14. The MIT Press, Cambridge, pp 689–695
15. Wainwright MJ, Jordan MI (2008) Graphical models, exponential families, and variational inference. Found Trends Mach Learn 1(1–2):1–305
16. Wainwright MJ, Jaakkola TS, Willsky AS (2003) Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudo-moment matching. In: Workshop on artificial intelligence and statistics
17. Beal MJ (2003) Variational algorithms for approximate Bayesian inference. Ph.D. thesis, University of Cambridge
18. Beal MJ, Ghahramani Z (2004) Variational Bayesian learning of directed graphical models with hidden variables. Bayesian Anal 1:1–44
19. Beal MJ, Ghahramani Z (2003) The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. In: Bernardo JM, Dawid AP, Berger JO, West M, Heckerman D, Bayarri MJ (eds) Bayesian statistics, vol 7. Oxford University Press, Oxford, pp 453–464
20. Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK (1999) An introduction to variational methods for graphical models. Mach Learn 37:183–233
21. Minka T (2005) Divergence measures and message passing. Technical report. MSR-TR-2005-173, Microsoft Research Ltd, Cambridge, UK
22. Bishop CM (2006) Pattern recognition and machine learning. Springer, Berlin
23. UIML Repository. http://archive.ics.uci.edu/ml/