Expert Systems with Applications 36 (2009) 3982–3989

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Adaptive spherical Gaussian kernel in sparse Bayesian learning framework for nonlinear regression

Jin Yuan [a,d], Liefeng Bo [b], Kesheng Wang [c,*], Tao Yu [a]

[a] CIMS and Robot Center, Shanghai University, 200072 Shanghai, China
[b] Institute of Intelligent Information Processing, Xidian University, 710071 Xi'an, China
[c] Department of Production and Quality Engineering, Norwegian University of Science and Technology, N-7491 Trondheim, Norway
[d] School of Mechanical and Electronic Engineering, Shandong Agricultural University, 271018 Tai'an, China

ARTICLE INFO

Keywords:
Relevance vector machine (RVM)
Gaussian kernel function
Regression
Gradient descent algorithm
Bayesian evidence

ABSTRACT

Kernel based machine learning techniques have been widely used to tackle problems of function approximation and regression estimation. Relevance vector machine (RVM) has state of the art performance in sparse regression. As a popular and competent kernel function in machine learning, conventional Gaussian kernel has unified kernel width with each of basis functions, which make impliedly a basic assumption: the response is represented below certain frequency and the noise is represented above such certain frequency. However, in many case, this assumption does not hold. To overcome this limitation, a novel adaptive spherical Gaussian kernel is utilized for nonlinear regression, and the stagewise optimization algorithm for maximizing Bayesian evidence in sparse Bayesian learning framework is proposed for model selection. Extensive empirical study, on two artificial datasets and two real-world benchmark datasets, shows its effectiveness and flexibility of model on representing regression problem with higher levels of sparsity and better performance than classical RVM. The attractive ability of this approach is to automatically choose the right kernel widths locally fitting RVs from the training dataset, which could keep right level smoothing at each scale of signal.

1. Introduction

Recently, under the statistical learning theory (Vapnik, 1998), kernel method has been widely used to tackle problems of function approximation and regression estimation. Some popular kernel regressions are the support vector machine (SVM) (Vapnik, 1998), Gaussian process (GP) (Williams & Rasmussen, 1996), relevance vector machine (RVM) (Tipping, 2000). The SVM was proposed for a regression estimation minimizing the norm of weight and loss function, which were developed originally for classification (Vapnik, 1998). SVM has delivered good performance in various applications. Gaussian process regression (GPR) is a Bayesian approach which assumes that target function is Gaussian process prior and using Bayesian inference acquires the prediction of unseen data (Rasmussen & Williams, 2006).

However, the SVM has a number of the significant and practical limitations (Tipping, 2001), for example, predictions are not probabilistic and the kernel function must satisfy Mercer's condition; that is, it must be a positive definite continuous symmetric function. It is also necessary to estimate the error/margin tradeoff parameter C (Smola & Schölkopf, 2004). GPR also is a probabilistic approach but sparse solution, the predictive distribution can be interpreted as a linear combination of *N* kernel functions, where *N* refers to the number of train data.

Tipping (2001) proposed a promising relevance vector machine, which has shown a comparable generalization performance but rather sparse solution than SVM. Although some approaches exploit the sparsity of SVM Smola and Schölkopf (2000) and the sparsity of GPR (Bo, Wang, & Jiao, 2006b; Csato & Opper, 2002), RVM is a general Bayesian learning framework of kernel method for obtaining state of the art sparse solutions to regression and classification tasks, which lead to significant reduction in the expense of computational complexity of the decision function and memory consumption of reconstructed predictive model, thereby making it more suitable for some real time applications (Agarwal & Triggs, 2004; Williams, Blake, & Cipolla, 2005). Note that RVM infers hyperparameters effectively by maximizing the marginal likelihood instead of time-consuming cross validation for model selection. In addition, the number of support vectors is sensitive to given error bound and grows linearly with the size of the training set, while the number of relevance vectors of RVM keeps sparseness and stability after the relevance vectors have ability to describe the distribution of the problem (Yuan, Wang, Yu, & Fang, 2007).

[*] Corresponding author.
E-mail addresses: jinyuan72@yahoo.cn (J. Yuan), kesheng.wang@ntnu.no (K. Wang).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.02.055

The choice of an appropriate kernel function is critical in order to obtain good generalization performance. Gaussian kernel is a very popular and competent kernel function in machine learning. In fact, Gaussian kernel is a local kernel which just responses to near neighbor of input variables and its local characteristic can be regarded as a multidimensional filter. Conventional Gaussian kernel k_0 (2) has unified kernel width with each basis function, which makes impliedly, in signal processing perspective, a basic assumption: the response is represented below certain frequency and the noise is represented above such certain frequency. However, in many case, this assumption does not hold. Under this assumption, for the signal contained large range of frequencies, the regression would lead to severe overfitting or oversmoothing even both at the same time (Fig. 3). Schmolck and Everson (2007) present an enforcing sparsity constraints scheme to control sparsity by incorporating a flexible noise-dependent smoothness prior into RVM. In this paper, the adaptive kernel width of Gaussian kernel k_2 (4) is utilized to cure the problem in such situation by tuning the spherical Gaussian kernel width to fit the local signal in different resolution scale.

To overcome this limitation, this paper combines the two paradigms: the adaptive kernel width of Gaussian kernels and sparse RVM learning, and presents an adaptive Gaussian kernel under Bayesian learning framework for relevance vector machines regression. This approach also could be adapted to classification, but this paper emphasizes on regression problem.

The rest of this paper is organized as follows: Section 2 analyzes the different type Gaussian kernel in kernel methods. The regression model of RVM is introduced concisely in Section 3. Section 4 describes the stagewise optimization of marginal likelihood maximization to infer hyperparameters and tune the adaptive kernel width of Gaussian kernel. And the empirical study on simulation experiments and discussion are presented in Section 5. The conclusion is drawn finally.

2. Gaussian kernel functions in kernel methods

Kernel approaches control the regression model with loss function, kernel function, and additional capacity or complexity control. The kernel function is used to construct a nonlinear response hyper-surface on the input space. In statistical method, kernel function offers an alternative solution by mapping data into high dimensional feature space to increase computational power (Muller, Ratsch, & Scholkopf, 2001). Typically, a regression model would be the linearly-weighted sum of kernel functions:

$$y(X) = \sum_{m=1}^{M} \omega_m k(X_m, X) \tag{1}$$

where $y(X)$ is an estimation at unseen data X, and M $(0 < M \leqslant N)$ is the number of support vectors (SVs) or relevance vectors (RVs). The kernel function $k(X_m, X)$ conducts the similarity measurement between SV (or RV) X_m and vector X in input space. Note that the kernels k must be the positive semi-definite kernel.

Generally, in the machine learning practice, using Gaussian kernel, also known as radial basis function (RBF) kernel, will yield better prediction performance (Smola & Schölkopf, 2004). A popular spherical Gaussian kernel k_0 takes the form:

$$k_0(X_m, X) = \exp\left(-\sum_{d=1}^{D} \frac{\|x_m^{(d)} - x^{(d)}\|^2}{2\ell^2}\right) \tag{2}$$

where ℓ is unified kernel width. Note that the width ℓ is invariant to the features of input space $\mathbb{R}^D$ and the SVs (or RVs), which means all of sampling vectors and its feature using constant width. Fig. 1 shows the one dimensional *Sinc* toy example of regression, which is linear combination of Gaussian kernel functions with a unified width. Therefore, in the domain where contains both high and low frequencies of response variation, the learning would be underfitting in the subdomain of high frequencies, and while the learning would suffer from overfitting in the subdomain of low frequencies if using a larger unified width of Gaussian kernel.

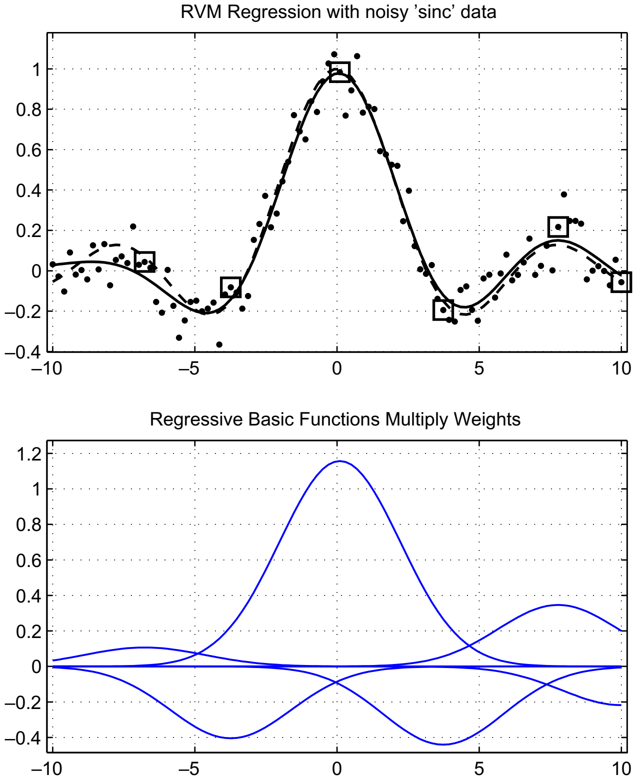

Fig. 1. Noisy *Sinc* RVM regression and weighted of Gaussian kernel basis function with global kernel width $\ell = 3$.

So, with extension of features, the elliptical Gaussian kernel k_1 takes the form:

$$k_1(X_m, X) = \exp\left(-\sum_{d=1}^{D} \frac{\|x_m^{(d)} - x^{(d)}\|^2}{2\ell_d^2}\right) \tag{3}$$

where $\ell_d(0 < d < D)$ is the feature scaling factor. Note that the width is variant to features of the input space $\mathbb{R}^D$ but not to the number of SVs (or RVs) M, which used in Automatic Relevance Determination (ARD) (Rasmussen & Williams, 2006; Tipping, 2001) or feature scaling methods (Bo, Wang, & Jiao, 2006a; Chapelle, Vapnik, Bousquet, & Mukherjee, 2002). If some features are unimportant or irrelevant for regression, the associated feature scaling factor will be small; otherwise it will be large.

Sequentially, generalizing the formulation of features to SVs (or RVs), the adaptive spherical Gaussian kernel k_2 takes the form:

$$k_2(X_m, X) = \exp\left(-\sum_{d=1}^{D} \frac{\|x_m^{(d)} - x^{(d)}\|^2}{2\ell_m^2}\right) \tag{4}$$

where $\ell_m(0 < m < M)$ is the adapting width factor. Note that the width is variant to different SVs (or RVs). In fact, this idea has been used to construct the centers and variances of radial basis function neural network (RBFNN), but not seen in the literatures of SVM or RVM. If the local response varies drastically, the associated adapting factor should be small; otherwise it should be large. Therefore, a learning mechanism could adjust the kernels width to adapt the variation of response. Theoretically, all the regression based on kernel methods could be optional, however, it should have as sparse as possible SVs (or RVs) to keep high efficiency of learning process, which RVM has state of the art sparse approach therefore leading to predictors of choice.

An extension of adaptive elliptical Gaussian kernel k_3 (5), which has introduced both feature scaling factor and adapting width factor, has more model flexibility with SVs or RVs Set, however, there are also much more parameters to tune, thus lead to the huge computational expense and easier overfitting (Cawley & Talbot, 2007). In general, it is inapplicable and beyond this discussion.

$$k_3(X_m, X) = \exp\left(-\sum_{d=1}^{D} \frac{\|x_m^{(d)} - x^{(d)}\|^2}{2\ell_{md}^2}\right) \tag{5}$$

3. Relevance vector machine

Relevance vector machine (Tipping, 2001) is simply a specialization of a spares Bayesian model which utilizes the same data-dependent kernel basis. The key feature of RVM is that the inferred predictors are exceedingly sparse in that they contain relatively few "relevance vectors", as well as offering good generalization performance. For this self-contained paper, RVM for regression is introduced concisely here.

Supposing the mapping relationship is multiple-input-single-out (MISO), sampled a dataset of N input vectors $\{X_n\}_{n=1}^{N}$ along with N corresponding scalar-valued target $\{t_n\}_{n=1}^{N}$, and assuming that the outputs are independent, identically distributed (IID) observations. In the engineering view, for some observations could be assumed to contain mean-zero Gaussian noise with variance $\sigma^2 : p(\varepsilon_n|\sigma^2) = \mathcal{N}(0, \sigma^2)$.

$$t = y(X; W) + \varepsilon = \Phi W + \varepsilon \tag{6}$$

where $t = [t_1, \ldots, t_N]^{\mathrm{T}},\ W = [\omega_1, \ldots, \omega_M]^{\mathrm{T}}$ is the weight vector, and where Φ is the $N \times M$ design matrix, wherein its element is $\phi_{nm} = k(X_m, X_n)$. In fact, the sparse Bayesian learning framework has the ability to utilize arbitrary basis functions, such as Gaussian kernel, splines kernel, symmlet wavelet kernel, Haar wavelet kernel (Schmolck & Everson, 2007), etc.

The classical approach to estimating t is to maximize likelihood (7) or to minimizing "least-squares" of the measured training dataset to estimate of W and σ^2, however, it would lead to over-fitting (Tipping, 2004).

$$p(t|W, \sigma^2) = (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left\{-\frac{1}{2\sigma^2}\|t - \Phi W\|^2\right\} \tag{7}$$

To control the complexity of model and avoid over-fitting, a zero-mean Gaussian prior probability distribution is defined over every ω_i with variance σ_i^{-1}, the prior of W is written as:

$$p(W|\alpha) = (2\pi)^{-M/2} \prod_{m=1}^{M} \alpha_m^{1/2} \exp\left\{-\frac{\alpha_m \omega_m^2}{2}\right\} \tag{8}$$

where hyperparameters vector $\alpha = [\alpha_0, \alpha_1, \alpha_2, \cdots \alpha_N]^{\mathrm{T}}$ controls how far from zero each weight is allowed to deviate. For completion of hierarchical prior, hyperpriors over $\alpha : p(\alpha)$ and the inverse noise variance $\sigma^2 : p(\sigma^2)$ are specified as Gamma distributions (9), within a, b, c, d sets to some uninformative value (*e.g.*, $a = b = c = d = 10^{-4}$).

$$\begin{aligned} p(\alpha) &= \Gamma(\alpha|a, b) = \prod_{n=0}^{N} \frac{b^a}{\Gamma(a)} \alpha^{a-1} \mathrm{e}^{-b\alpha} \\ p(\sigma^2) &= \Gamma(\sigma^2|c, d) = \frac{d^c}{\Gamma(c)} \sigma^{-2(c+1)} \mathrm{e}^{-b/\sigma^2} \end{aligned} \tag{9}$$

Consequently, using Bayes' posterior inference, the posterior distribution over W is also conveniently Gaussian: $p(W|t, \alpha, \sigma^2) \sim \mathcal{N}(\mu, \Sigma)$. Where the posterior mean μ and covariance Σ are as follows:

$$\mu = \sigma^{-2} \Sigma \Phi^{\mathrm{T}} t \tag{10}$$

$$\Sigma = (\sigma^{-2} \Phi^{\mathrm{T}} \Phi + A)^{-1} \tag{11}$$

where $A = \mathrm{diag}(\alpha_0, \alpha_1, \cdots \alpha_N)$.

Sparse Bayesian learning can then be formulated as maximization of hyperparameter posterior, $p(\alpha, \sigma^2|t) \propto p(t|\alpha, \sigma^2)p(\alpha)p(\sigma^2)$. In practice, due to $\alpha, \sigma^2 > 0$, to avoid adding positive constraints in the optimization problem, their logarithm are considered, then uniform hyperpriors are defined over a logarithmic scale, which ultimately raises spare solutions (Quiñonero Candela, 2004). So the MAP of hyperparameter need only to maximize the marginal likelihood $p(t|\alpha, \sigma^2)$, known as type-II Maximum Likelihood procedure. The RVM marginal likelihood, also called *evidence* by MacKay (1992), is given by:

$$\begin{aligned} L(\alpha, \sigma^2) = p(t|\alpha, \sigma^2) &= \int p(t|X, W, \sigma^2)p(W|\alpha)\,\mathrm{d}W \sim N(0, C) \\ &= -\frac{1}{2}\left[N \log 2\pi + \log|C| + t^{\mathrm{T}} C^{-1} t\right] \end{aligned} \tag{12}$$

where the covariance is $C = \sigma^2 I + \Phi A^{-1} \Phi^{\mathrm{T}}$.

Obtained in this optimization process, the value of α_{MP} and σ_{MP}^2, as the substitution of the α and σ^2, the posterior mean (9) and variance (10) can be computed, and then a mean final approximator at unseen data X^* could be gained with:

$$\mu_* = \mu^{\mathrm{T}} \Phi(X^*) \tag{13}$$

If the sparse Bayesian learning framework utilizes Gaussian kernel $k_0(X_m, X)$ basis function, cross-validation on the validation set is used to get good unified kernel width.

4. Stagewise optimization for evidence maximization with adaptive width Gaussian kernel

The extension of classical RVM in Tipping (2001) focuses on modify the type and number of basis function and directly optimize the evidence with respect to the kernel parameters. Inspired by this sparse Bayesian learning algorithm which used feature scaling Gaussian kernel k_1 for dealing with irrelevant input features, this paper shares same basic learning framework but applying it to the RVM with adaptive width Gaussian kernel k_2.

In this section, as stagewise optimization method, we first deal with the *evidence* maximization with respect to α and σ^2. And then the *evidence* with respect to ℓ is maximized using gradient descent algorithm in second optimization stage. Sequentially, the algorithm is presented in Section 4.3. Finally the future direction is discussed.

4.1. Evidence maximization with respect to α and σ^2

Tipping (2001) suggested two type of optimization techniques on the MAP of hyperparameter: minimization the negative log *evidence* using gradient descent algorithm and maximization positive log *evidence* using expectation maximization (EM) procedure.

In practice, gradient descent algorithm scheme minimizes the negative log *evidence* $\mathscr{L}$ with respect to the hyperparameters $\log\alpha$ and $\log\sigma^2$. Dropping the constant term, it is given by:

$$\begin{aligned} \mathscr{L}(\log\alpha, \log\sigma^2) &= \frac{1}{2}\left[\log|\sigma^2\mathbf{I} + \Phi\mathbf{A}^{-1}\Phi^{\mathbf{T}}| + t^{\mathrm{T}}(\sigma^2\mathbf{I} + \Phi\mathbf{A}^{-1}\Phi^{\mathbf{T}})^{-1} t\right] \\ &= \frac{1}{2}\left[\sigma^{-2} t^{\mathrm{T}}(t - \Phi\mu) + N\log\sigma^2 - \log|\Sigma| - \log|A|\right] \end{aligned} \tag{14}$$

The derivatives of $\mathscr{L}$ with respect to $\log\alpha$ and $\log\sigma^2$ hold:

$$\frac{\partial\mathscr{L}}{\partial\log\alpha_m} = -\frac{1}{2} + \frac{1}{2}\alpha_m(\mu_m^2 + \Sigma_{mm}) \tag{15}$$

$$\frac{\partial\mathscr{L}}{\partial\log\sigma^2} = \frac{N}{2} - \frac{1}{2\sigma^2}(tr[\Sigma\Phi^{\mathrm{T}}\Phi] + \|t - \Phi\mu\|^2) \tag{16}$$

Set the derivatives to zero and obtain the iterative form:

$$\alpha_m^{\text{new}} = \frac{1}{\mu_m^2 + \Sigma_{mm}} \tag{17}$$

$$(\sigma^2)^{\text{new}} = \frac{\|t - \Phi\mu\|^2 + tr[\Sigma\Phi^{\mathrm{T}}\Phi]}{N} \tag{18}$$

Another scheme, the EM algorithm proceeds a lower bound $\mathscr{F}(q, \alpha, \sigma^2)$ on positive log evidence by defining a variational probability distribution $q(W)$. So in the E-step $\mathscr{F}$ is maximized with respect to $q(W)$ for fixed parameters α and σ^2, and in M-step $\mathscr{F}$ is maximized with respect to $\alpha$and σ^2 for fixed $q(W)$. Since the posterior over W is Gaussian, the E-step need only to compute μ (10) and Σ (11). In M-step, dropping the constant term, $\mathscr{F}$ is rewritten and maximized:

$$\int q(W)\log p(t, W|\alpha, \sigma^2)\,\mathrm{d}W = \frac{1}{2}\log|A| - \frac{1}{2}\mathrm{tr}[A\Sigma + A\mu\mu^{\mathrm{T}}] - \frac{N}{2} \times \log\sigma^2 - \frac{1}{2\sigma^2}(\|t - \Phi\mu\|^2 + \mathrm{tr}[\Sigma\Phi^{\mathrm{T}}\Phi]) \tag{19}$$

Take derivatives with respect to α and σ^2, and set them to zero, the result equivalent to gradient descent scheme. More detail on EM learning for the RVM refers (Quiñonero Candela, 2004).

By introducing the quantities $\gamma_m = 1 - \alpha_m\Sigma_{mm}$, which are a measure of how "well-determined" each w_m is by the train data (MacKay, 1992), the faster convergence update form holds:

$$\alpha_m^{\text{new}} = \frac{\gamma_m}{\mu_m^2} \tag{20}$$

$$(\sigma^2)^{\text{new}} = \frac{\|t - \Phi\mu\|^2}{N - \Sigma_m\gamma_m} \tag{21}$$

Fig. 2 shows the convergence rate comparison of different update algorithm. The MacKay update form has highly effective on pruning RVs.

4.2. Evidence maximization with respect to ℓ

In practice, the negative log evidence is minimized with respect to ℓ using gradient descent algorithm. To avoid adding positive constraints in the optimization problem, we use parameterizations $\log\ell$. So this is rewritten:

$$\mathop{\arg\min}_{\log\ell}\left(\mathscr{L} = \frac{1}{2}[\log|C| + t^{\mathrm{T}}C^{-1}t]\right) \tag{22}$$

where $C = \sigma^2\mathbf{I} + \Phi\mathbf{A}^{-1}\Phi^{\mathbf{T}}$.

According to the chain rule, the gradient of the evidence $\mathscr{L}$ with respect to the kernel width of mth $(m = 1, 2, \ldots, M)$ RVs is first written in the form:

$$\frac{\partial\mathscr{L}}{\partial\log\ell} = \frac{\partial\mathscr{L}}{\partial\Phi}\cdot\frac{\partial\Phi}{\partial\log\ell} \tag{23}$$

Due to the first term in (23) is independent of the kernel function parameters, so it follows that we need only to calculate $\frac{\partial\mathscr{L}}{\partial\Phi}$ and $\frac{\partial\Phi}{\partial\log\ell}$, respectively. Firstly, in the computation of matrix (Roweis, 1999), for the log determinant of a positive definite symmetric matrix C, $\frac{\partial}{\partial x}\log|\det(C)| = C^{-1}$, and for the derivative of symmetric inverse matrix C^{-1}, $[a^{\mathrm{T}}C^{-1}b]' = -C^{-1}ab^{\mathrm{T}}C^{-1}$ So we have

$$\frac{\partial\mathscr{L}}{\partial\Phi} = \frac{1}{2}\frac{\partial}{\partial\Phi}[\log|C| + t^{\mathrm{T}}C^{-1}t] = \frac{1}{2}(C^{-1} - C^{-1}tt^{\mathrm{T}}C^{-1}t)\frac{\partial C}{\partial\Phi} \tag{24}$$

And for the symmetric matrix A, the derivative of square form: $(X^{\mathrm{T}}AX)' = 2X^{\mathrm{T}}AX'$, then we have

$$\frac{\partial C}{\partial\Phi} = \frac{\partial}{\partial\Phi}(\sigma^2\mathbf{I} + \Phi\mathbf{A}^{-1}\Phi^{\mathbf{T}}) = 2\Phi\mathbf{A}^{-1} \tag{25}$$

Then substituting (25) to (24), The derivative of $\mathscr{L}$ with respect to Φ is given by

$$\frac{\partial\mathscr{L}}{\partial\Phi} = (C^{-1} - C^{-1}tt^{\mathrm{T}}C^{-1})\Phi A^{-1} \tag{26}$$

By defining $E_{nm} = \partial\mathscr{L}/\partial\phi_{nm}$, the form of (26) is rewritten as a more intuitive form (Tipping, 2001):

$$E = \sigma^2[\Phi\Sigma - (t - y)\mu^{\mathrm{T}}] \tag{27}$$

So for the set of Gaussian kernel functions $\phi_{nm} = k_2(X_m, X)$, the partial derivatives of $\mathscr{L}$ with respect to the logarithm of adaptive width parameters $(\gamma_m = 1/2\ell_m^2)$:

$$\frac{\partial\phi_{nm}}{\partial\log\gamma_m} = -\sum_{d=1}^{D}(x_m^{(d)} - x_n^{(d)})^2\cdot\phi_{nm} \tag{28}$$

Combining Eqs. (27) and (28), we can compute the derivatives (23) of the evidence $\mathscr{L}$:

$$\frac{\partial\mathscr{L}}{\partial\log\gamma_m} = \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{d=1}^{D} -E_{nm}\phi_{nm}(x_m^{(d)} - x_n^{(d)})^2 \tag{29}$$

4.3. The algorithm

The stagewise optimization of *evidence* maximization with respect to hyperparameters, based on the classical RVM algorithm, is described in Algorithm 1.

In the first stage, the hyperparameters α and σ^2 are optimized (line 3–5) while the kernel parameter ℓ is fixed. After iterating H cycles of the α and σ^2 optimization, with the fixed hyperparameters re-estimated α and σ^2, the kernel parameter ℓ is optimized in second stage (line 7–10). Because of the change of kernel parameter and then Φ, thus μ and Σ is updated (line 8) to reflect the current state of the model.

Algorithm 1. Stagewise optimization of RVM with adaptive width Gaussian kernel algorithm

```
1. Initialize $\alpha \leftarrow [1/N, \ldots, 1/N]^{\mathrm{T}}; \sigma^2 \leftarrow 0.1 \times \mathrm{var}(t)$ and $\ell$;
2. for $step_1 = 1$ to maximum iteration or convergence
3.     Compute $\{\mu, \Sigma, \Phi\}$;
4.     Re-estimate $\alpha_i^{\text{new}}$ and $(\sigma^2)^{\text{new}}$;
5.     Delete near-zero RVs;
6.     If $step_1 \bmod H = 0$ (e.g. H=5 or 10)
7.         for $step_2 = 1$ to maximum iteration (e.g. $S = 5$ or 10) or convergence
8.             Update $\{\mu, \Sigma, \Phi\}$;
9.             Using gradient descent algorithm to find a better $\ell$;
10.        end for
11.    end if
13. end for
14. Predicting for unseen data using the regressive model (13).
```

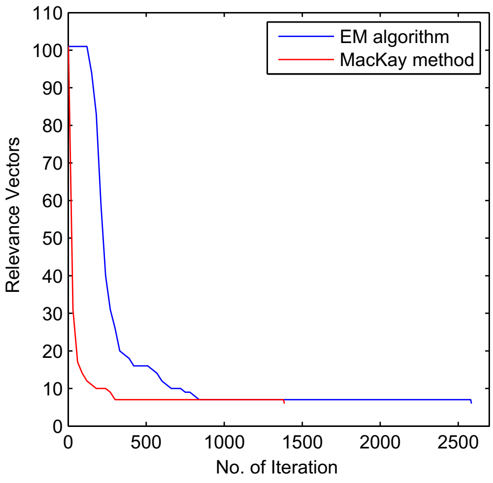

Fig. 2. Convergence rate comparison.

Following Tipping (2001), the convergence criterion in first stage (line 2) is set to the largest absolute change between sequential iterations in $\log \alpha_m < 10^{-9}$, and the convergence criterion in second stage (line 7) is set to that the log-evidence $\mathscr{L}$ changes by less than 10^{-6}. In practice, we have found it is effective to prevent premature convergence by setting limited optimization frequency H (line 6) and maximum iteration S in second stage.

As a function of α and σ^2, the negative log evidence has multiple minima (Quiñonero Candela, 2004), let alone being a function of α, σ^2 and ℓ, so it is intractable to converge to the global optimum with the greedy nature and stagewise update of the optimization technique. Therefore, the initial kernel parameter ℓ still has some effect on regressive performance, which could be improvement via grid research and cross-validation.

4.4. Future directions

The computational complexity of the algorithm is

$$\#(step_1) \times (N^3 + \#(step_2) \times M \times N^3/H) \tag{30}$$

where $step_1$ and $step_2$ refer to the maximum iteration in first stage and second stage. Why the above algorithm works applicable with so many hyperparameters and kernel parameters optimization? Its mainly reason is the highly effective pruning RVs from the N to M (typically, $M \ll N$). Obviously, the bottleneck of the algorithm is the computation of inverse operation (Cholesky decomposition) of $O(N^3)$ complexity. Although empirically the type-II Maximum Likelihood scheme on which we base this work does not seem to be much of an issue under most medium-size scenarios, it quickly becomes intractable as the number of training samples increases. In the future, an approach based on fast RVM algorithm (Tipping & Faul, 2003), RVM* algorithm (Quiñonero Candela, 2004) or fast generalized cross-validation algorithm (Sundararajan, Shevade, & Sathiya Keerthi, 2007) is an alternative approach for larger dataset.

5. Empirical study and discussion

In order to demonstrate the effectiveness of adaptive width Gaussian kernel k_2 RVM in sparse Bayesian learning framework, we compare its performance with those of RVM with classical Gaussian kernel k_0 in the regression experiments with 2 artificial datasets and 2 real-world benchmark datasets. All the algorithms are implemented in MATLAB 7.0.

5.1. Optimization implementation

For the stagewise optimization in sparse Bayesian learning framework, a gradient-descent method is used in second stage to search for the optimal values for the kernel parameters, and thus one needs to choose good optimization software. We recommend using an available optimization package to avoid the numerical problems. Here we use the function *fminunc* in the optimization toolbox of MATLAB that implements BFGS quasi-Newton algorithm to solve medium-scale problems. The maximum number of iterations allowed is set to be 10, the termination tolerance on the function value and variable value is set to be 10^{-5}, and the cubic polynomial line search procedure is used to find the optimal step size.

5.2. Multiscale data set

The multiscale resolution data task is to estimate a regression of a noisy function, given N examples (x_i, y_i):

$$y_i = \sin\left(\frac{40}{x_i}\right) + \varepsilon \tag{31}$$

where x_i drawn uniformly from $[1,10]$, and ε is drawn from a Gaussian distribution with mean 0 and variance σ^2. For the dataset, the training set consisted of 150 samples while the test set has 200 noise-free samples. the settings of this simulation are: $S = 10;\ H = 10;\ \sigma = 0.1$.

For comparison with this approach, Fig. 3 shows the multiscale resolution data regression with classical RVM using Gaussian kernel k_0 (2), which results in severe overfitting (Fig. 3a) or oversmoothing (Fig. 3c) even both at the same time (Fig. 3b), while the same regression problem with this approach using Gaussian kernel k_2 (4) trained by this stagewise approach shows its adaptive ability in multiscale situation in Fig. 4, which adaptively fit the response at each scale, but not the noise, while keeping right level of smoothing. The adaptive widths Gaussian kernel of RVs in Fig. 4 is shown in Fig. 5. Note that with same initial parameter ℓ the classical RVM approach (Fig. 4b) has more RVs than this approach (Fig. 5) in low-frequency subdomain of response, meanwhile, in high-frequency subdomain of response, low predictive performance with almost same RVs.

5.3. Tipping two dimensional Sinc data set

Another toy example sampled from the function, which was used to test the feature scaling problem in (Tipping, 2001):

$$y_i = \frac{\sin(x_i^{(1)})}{x_i^{(1)}} + 0.1x_i^{(2)} \tag{32}$$

y_i is corrupted by the Gaussian noise with mean 0 and variance σ^2, both dimensional x_i drawn uniformly from [−10, 10]. The simulation is conduct by the classical RVM approach with Gaussian kernel k_0 (k_0, for short) and stagewise optimization RVM approach with Gaussian kernel k_2 (k_2, for short). The comparison results are shown in Table 1, and the settings of this simulation are: $S = 10;\ H = 10;\ \sigma = 0.1$; The initial width set to 3. For the test set has 200 noise-free generated samples.

From the Table 1, we can see that this approach obtained more sparseness regression than classical RVM in all dataset size with better predictive performance. It is reasonable for the relevance vector kernels set using k_2 with kernel widths adapted by an optimization process more flexible than the kernels set using k_0. Obviously, the sparseness is affected by the training dataset size. The sparseness obtained by this approach is more obvious than the classical RVM when the dataset distributing with high density in the domain, such as dataset size from 15 * 15 to 30 * 30. Meanwhile, when dataset size increases to certain extent, after the RVs have the ability to describe the nonlinear process, the number of vectors of RVM keeps sparseness and stability. On the contrary, when the training dataset distributes insufficient data in low dimension input domain (*e.g.* dataset 8 * 8 and 10 * 10), the two approaches have almost similar sparseness. Regarding the convergence and sparsity procedure, the comparison between two approaches on 15 * 15 dataset is shown in Fig. 6. The second stage optimization maximized the *evidence* with respect to ℓ, shown by the step improvement in red in left panel, furthermore, affect the maximum evidence of whole optimization and the convergence speed, also more sparsity in right panel.

5.4. Benchmark data set

We tested the two approaches on two real-world datasets[1]: the Boston housing dataset (Harrison & Rubinfeld, 1978) and the Abalone dataset (Newman & Asuncion, 2007). The Boston housing data-

[1] The dataset is obtained from http://www.gatsby.ucl.ac.uk/~chuwei/regression.html.

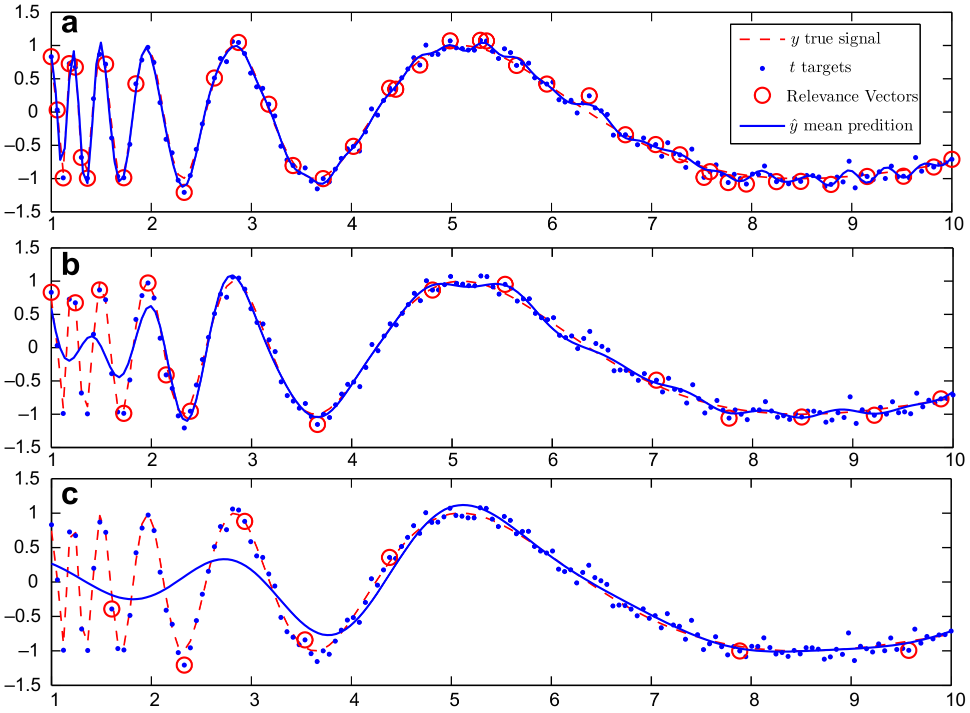

Fig. 3. Multiscale resolution data regression by classical RVM with global width (a) $\ell = 0.2$, (b) $\ell = 0.41$ and (c) $\ell = 1.2$.

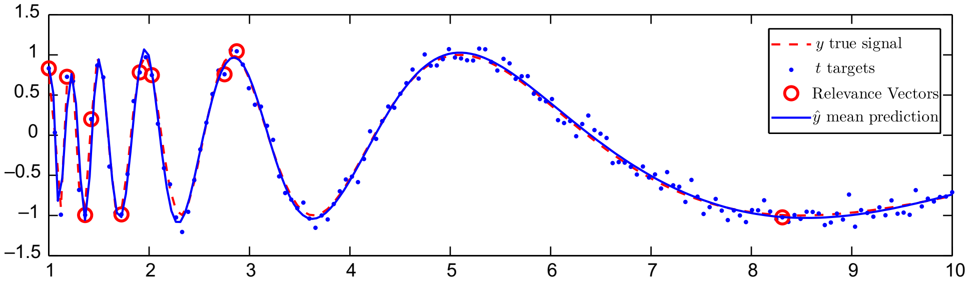

Fig. 4. Multiscale resolution data regression by stagewise optimized RVM with adapting width initialized $\ell = 0.41$, regression test error (RMS): 0.0671.

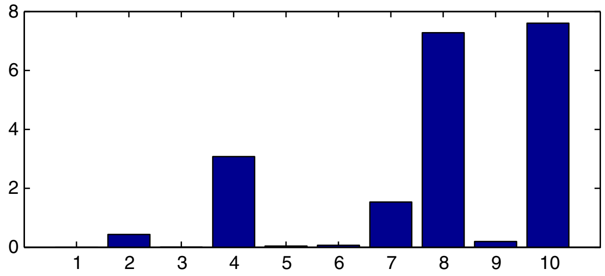

Fig. 5. The adapting width factors corresponding to RVs with same sequence in Fig. 4.

set contains 506 instances with 13 features, which split into 481 instances for training and 25 for test in the experiment. The Abalone dataset comprises 4177 patterns with eight attributes. All the patterns are normalized to zero mean and unit variance coordinate-wise, and randomly partitioned in 1000 patterns for training and 3177 patterns for test. The settings of this simulation are: $H = 15$; $S = 8$; the initial width set to 5.5. The comparison results

Table 1
Sparsity and prediction performance comparison on tipping two dimensional *Sinc* data

Training dataset size	Methods	RMS (10^{-2})	RVs
8 * 8	Classical RVM	16.75	63
	Proposed method	14.15	59
10 * 10	Classical RVM	6.35	99
	Proposed method	6.21	98
15 * 15	Classical RVM	6.38	104
	Proposed method	6.15	89
20 * 20	Classical RVM	5.49	115
	Proposed method	5.46	90
25 * 25	Classical RVM	5.43	122
	Proposed method	5.42	91
30 * 30	Classical RVM	5.36	128
	Proposed method	5.32	93

are shown in Table 2. From the comparison, proposed approach also obtained sparser RVs set than classical RVM with better prediction performance.

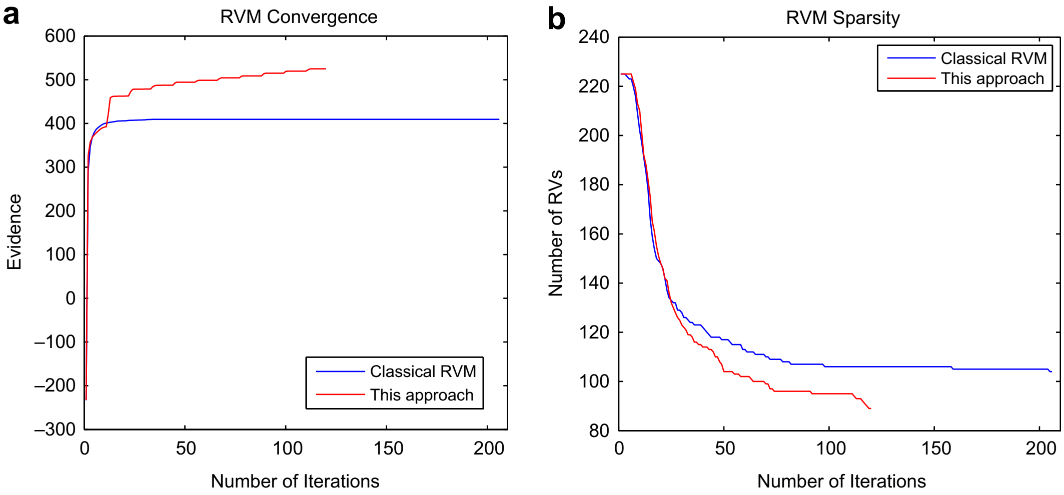

Fig. 6. The comparison of RVM convergence and sparsity procedure for the 15 * 15 dataset of tipping two dimensional *Sinc* data.

Table 2
Prediction performance and sparsity comparison on benchmark data

Dataset	Methods	MSE	RVs
Boston housing	Classical RVM	8.21959	39
	Proposed method	8.05299	31
Abalone	Classical RVM	0.46723	21
	Proposed method	0.45220	18

5.5. Discussion

We explore the test error (RMS) with respect to unified kernel width on multiscale training dataset shown in Fig. 7a. The minimum test error (0.0705 with kernel width 0.16, but 48 RVs) represents a learning ability of classical RVM with Gaussian kernel k_0. However, compared with the result (0.0671 and only 10 RVs in Fig. 4), the deficiency of learning ability of classical RVM mainly because the absence of local width tuning capability in nature as a kind of local kernel, especially in the situation of containing varied frequency signal. So, the attractive ability of this approach is to automatically choose the right kernel widths locally fitting RVs from the training dataset.

However, to add the flexibility of the kernel function appears to call for an effective way to deal with local maxima. When the training dataset distributes highly sparseness in high dimension input domain, (*e.g.* the Boston housing dataset), the model with many hyperparameter incurs many local maxima. If in this situation, the resulting in performance will be sensitive to the initialization of kernel width.

6. Conclusions

In this paper, as a straightforward extension to the RVM, a novel use of adaptive spherical Gaussian kernel is proposed for nonlinear regression, and we have described the stagewise optimization algorithm for maximizing marginal likelihood, also known as Bayesian evidence, of the model in sparse Bayesian learning framework. In the first stage, the hyperparameters α and σ^2 are tuned by maximizing evidence with effective MacKay update methods, which could quickly make the model sparseness. In the second stage, the kernel parameters linked with RVs is adapted by gradient descent algorithm to maximize evidence. The attractive ability of this approach is to automatically choose the right kernel widths locally fitting RVs from the training dataset.

Compared with classical RVM with unified Gaussian kernel width, the regression experiments, two artificial datasets and two real-world benchmark datasets, show that adaptive spherical Gaussian kernel RVM with stagewise optimization is effectiveness and flexibility of model on representing regression problem with higher performance and higher levels of sparsity. It will also be

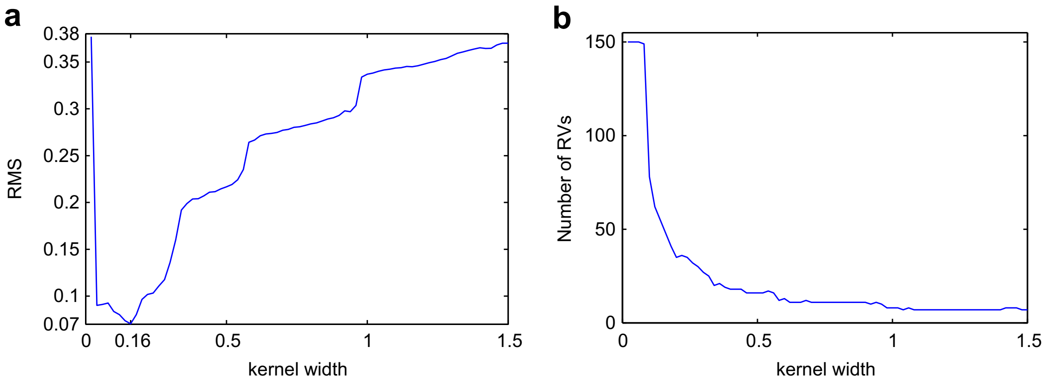

Fig. 7. Prediction performance and sparsity of classical RVM on multiscale dataset with a range of kernel width parameter.

interesting and easy to extend the proposed algorithm to classification problems.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (NSFC) under the Grant No. 50375090.

References

Agarwal, A., & Triggs, B. (2004). 3D human pose from silhouettes by relevance vector regression. In *Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition* (Vol. 882, pp. II-882–II-888).

Bo, L., Wang, L., & Jiao, L. (2006a). Feature scaling for kernel fisher discriminant analysis using leave-one-out cross validation. *Neural Computation, 18*(4), 961–978.

Bo, L. F., Wang, L., & Jiao, L. C. (2006b). Sparse gaussian processes using backward elimination. *Advances in Neural Networks, 3971*(1), 1083–1088.

Cawley, G. C., & Talbot, N. L. C. (2007). Preventing over-fitting during model selection via bayesian regularisation of the hyper-parameters. *Journal of Machine Learning Research, 8*, 841–861.

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning, 46*(1/3), 131.

Csato, L., & Opper, M. (2002). Sparse on-line gaussian processes. *Neural Computation, 14*(3), 641–668.

Harrison, D., & Rubinfeld, D. L. (1978). Hedonic prices and the demand for clean air.

MacKay, D. (1992). The evidence framework applied to classification networks. *Neural Computation, 4*(5), 720–736.

Muller, M., Ratsch, T., & Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEETNN: IEEE Transactions on Neural Networks, 12*.

Newman, D. J., & Asuncion, A. (2007). UCI machine learning repository. Available from http://www.ics.uci.edu/~mlearn/MLRepository.html.

Quiñonero Candela, J. (2004). Learning with uncertainty – Gaussian processes and relevance vector machines. PhD thesis, Technical University of Denmark, Lyngby, Denmark.

Rasmussen, C. E., & Williams, C. (2006). *Gaussian processes for machine learning*. MIT Press.

Roweis, S. (1999). Matrix identities. Available from http://www.cs.toronto.edu/~roweis/notes/matrixid.pdf.

Schmolck, A., & Everson, R. (2007). Smooth relevance vector machine: A smoothness prior extension of the RVM. *Machine Learning, 68*(2), 107–135.

Smola, A., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing, 14*(3), 199–222.

Smola, A., & Schölkopf, B. (2000). Sparse greedy matrix approximation for machine learning. In P. Langley (Ed.), *Proceedings of the 17th international conference on machine learning* (pp. 911–918). San Francisco: Morgan Kaufmann.

Sundararajan, S., Shevade, S. K., & Sathiya Keerthi, S. (2007). Fast generalized cross-validation algorithm for sparse model learning. *Neural Computation, 19*(1), 283–301.

Tipping, M. (2000). The relevance vector machine. In A. Solla, T. Leen, & K.-R. Mller (Eds.). *Advances in neural information processing systems* (Vol. 12, pp. 652–658). Cambridge: MIT.

Tipping, M. (2004). Bayesian inference: An introduction to principles and practice in machine learning. *Advanced Lectures on Machine Learning, 3176*, 41–62.

Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research, 1*, 211–244.

Tipping, M. E. & Faul, A. C. (2003). Fast marginal likelihood maximization for sparse bayesian models. In *Proceedings of the ninth international workshop on artificial intelligence and statistics*.

Vapnik, V. (1998). *Statistical learning theory*. Wiley.

Williams, O., Blake, A., & Cipolla, R. (2005). Sparse Bayesian learning for efficient visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(8), 1292–1304.

Williams, C., & Rasmussen, C. (1996). Gaussian processes for regression. *Advances in Neural Information Processing Systems, 8*, 598–604.

Yuan, J., Wang, K., Yu, T., & Fang, M. (2007). Integrating relevance vector machines and genetic algorithms for optimization of seed-separating process. *Engineering Applications of Artificial Intelligence, 20*(7), 970–979.