Ann Oper Res (2018) 263:21–43
https://doi.org/10.1007/s10479-015-1890-9

Robust relevance vector machine for classification with variational inference

Sangheum Hwang[1] · **Myong K. Jeong**[2]

Published online: 30 August 2015

Abstract The relevance vector machine (RVM) is a widely employed statistical method for classification, which provides probability outputs and a sparse solution. However, the RVM can be very sensitive to outliers far from the decision boundary which discriminates between two classes. In this paper, we propose the robust RVM based on a weighting scheme, which is insensitive to outliers and simultaneously maintains the advantages of the original RVM. Given a prior distribution of weights, weight values are determined in a probabilistic way and computed automatically during training. Our theoretical result indicates that the influences of outliers are bounded through the probabilistic weights. Also, a guideline for determining hyperparameters governing a prior is discussed. The experimental results from synthetic and real data sets show that the proposed method performs consistently better than the RVM if a training data set is contaminated by outliers.

Keywords Relevance vector machine · Outlier · Robust classification · Sparsity

1 Introduction

The relevance vector machine (RVM) is a sparse kernel-based learning algorithm for regression and classification (Tipping 2001). In classification, the RVM represents a Bayesian treatment of the logistic regression model with independent priors over model coefficients governed by a set of hyperparameters. Specifically, an independent zero-mean Gaussian prior is assumed for each of model coefficients and an independent Gamma hyper prior is used for each hyperparameter. Then posterior distributions of model coefficients and hyperparameters are estimated from a training data set. Originally, the posterior distributions are

✉ Myong K. Jeong
mjeong@rci.rutgers.edu

[1] Department of Industrial and Systems Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea

[2] RUTCOR (Rutgers Center for Operations Research), Rutgers, The State University of New Jersey, Piscataway, NJ, USA

approximated by the *evidence procedure* known as the *type-II maximum likelihood* method (Mackay 1992). An alternative approach for the approximation is a variational inference method, which maximizes a variational lower bound on the marginal log likelihood (Bishop and Tipping 2000).

From the hierarchical prior structure referred as *automatic relevance determination* prior (Neal 1996), the posterior distributions of many of the model coefficients are sharply peaked around zero, and therefore, those coefficients can be eliminated from the final model. Consequently, we can obtain a sparse solution. The training observations with the non-zero coefficient values are called the *relevance vectors*. Another popular kernel-based learning algorithm which also provides a sparse solution is the support vector machine (SVM) (Burges 1998; Lee et al. 2014). In the SVM, the observations contributing to the resulting decision boundary are called the *support vectors*. Compared to the SVM, the RVM has several advantages in practice. First, a higher degree of sparsity can be obtained, i.e. the number of relevance vectors is much smaller than the number of support vectors. Second, it provides probabilistic outputs (e.g., class probability estimates). Finally, it is possible to control model complexity automatically, that is, there is no an additional parameter for the regularization.

However, if a data set contains outlying observations called outliers, a decision boundary from the RVM may be seriously distorted. Since the data set with outliers is frequently encountered in practice, it is desired to develop a robust learning algorithm for the RVM which is insensitive to the outliers. A simulated example in Fig. 1 illustrates the effect of an outlier (in a square box) on the decision boundaries from the SVM, RVM and the proposed method, which is called the robust RVM (RRVM), in this study. Figure 1a, b represents the decision boundaries obtained by employing the linear kernel and the radial basis function (RBF) kernel with $\sigma = 2$, respectively. For the SVM, the regularization constant C is set to 1. From the figure, it is observed that the decision boundaries from the SVM and RVM are pulled toward to the outlier regardless of the type of kernels; in contrast, the decision boundary of the proposed RRVM is less sensitive to the outlier (see Fig. 3 that illustrates how the decision boundary of RRVM is changed at each iteration).

There exist several works to develop robust kernel-based learning algorithms (Song et al. 2002; Hwang et al. 2014, 2015; Wu and Liu 2007; Debruyne et al. 2009; Park and Liu 2011). Wu and Liu (2007) proposed the robust truncated hinge loss SVM. Since the associated optimization problem involves nonconvex minimization, they applied the difference convex algorithm (An and Tao 1997) to solve the nonconvex problem through a sequence of convex sub-problems. Debruyne et al. (2009) utilized a weighting strategy based on their proposed spatial rank measure. If a specific observation has a lower spatial rank than a predefined threshold, the corresponding weight becomes zero. Therefore, observations which have a low spatial rank value are pruned from the training data set. However, these studies cannot provide any statistical information such as a class probability since they were developed based on the SVM approach. In contrast, Park and Liu (2011) considered a truncated logistic loss function for the logistic regression to eliminate the effect of outlying observations. Although this work is able to estimate the class probability, it does not give the sparse solution.

In this paper, we develop a robust learning algorithm based on the RVM, which is insensitive to outliers and simultaneously maintains the advantages of the RVM (e.g., class probability, sparsity, etc.). For this, a weighted logistic regression is considered to reduce the effect of outliers. The weights should be determined carefully since their incorrect estimates may lead to poor generalization performance. To determine the weights probabilistically, a fully Bayesian approach for the RRVM is considered combing with the variational inference method (Jaakkola 2000; Ormerod and Wand 2010) to obtain posterior distributions over all

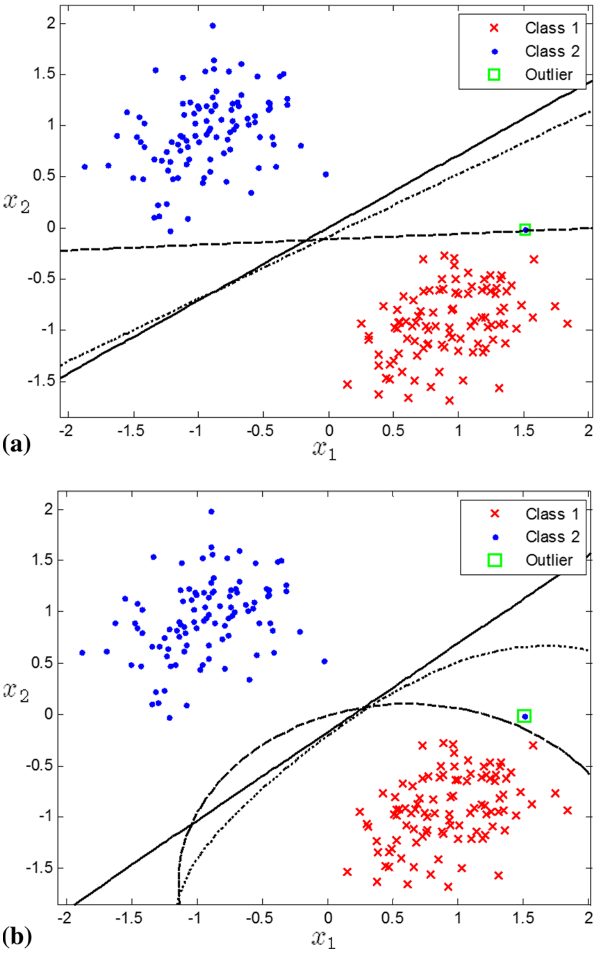


Fig. 1 A simulated example: plots of the decision boundaries from SVM (*dotted line*), RVM (*dashed line*) and RRVM (*full line*) for the **a** linear kernel and **b** RBF kernel with $\sigma = 2$

random variables. Through the incorporation of a variational inference method in a RRVM, we can avoid an extensive grid search such as cross validation to optimize the parameters associated with a kernel function and does not need an extra computational effort to obtain a robust estimator compared to the RVM derived from the variational inference (Bishop and Tipping 2000). Also, it is shown that the probabilistic weights in the RRVM can be interpreted as a bounded loss function which makes the effect of outliers bounded. The technical property of the weight value is also derived.

The rest of this paper is organized as follows. In Sect. 2, we briefly describe a standard logistic regression and its weighted version for achieving the robustness. In Sect. 3, the RRVM is developed incorporating with the variational inference and some theoretical properties of the RRVM are derived. The robustness of the proposed method is compared with common classification algorithms using synthetic and real data sets in Sect. 4. Finally, concluding remarks and future works are presented in Sect. 5.

2 Weighted logistic regression and regularization

Consider a data set of N input-target pairs $\{\mathbf{x}_i, t_i\}_{i=1}^{N}$, where $\mathbf{x}_i$ represents a d-dimensional input vector and t_i represents class labels: $t_i = 0$ if the $i-$th observation belongs to the first class and $t_i = 1$ if it belongs to the second class. A decision boundary can be defined as a linear combination of M basis functions as follows:

$$f\left(\boldsymbol{\phi}(\mathbf{x})\right) = \boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}) = \beta_0 + \sum_{i=1}^{M} \beta_i \phi_i(\mathbf{x})$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_M)^T$ is a vector of model coefficients and $\boldsymbol{\phi}(\mathbf{x}) = (1, \phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x}))^T$ is a vector of basis functions. By employing some nonlinear basis functions, the decision boundary $f\left(\boldsymbol{\phi}(\mathbf{x})\right)$ becomes a nonlinear function with respect to $\mathbf{x}$. Some commonly used basis functions are the polynomial kernel, $\phi_i(\mathbf{x}) = (1 + \langle \mathbf{x}, \mathbf{x}_i \rangle)^d$, where the parameter d is the degree of polynomial to be used, and the Gaussian RBF kernel,

$$\phi_i(\mathbf{x}) = \exp\left\{ -\frac{(\mathbf{x} - \mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i)}{\sigma} \right\},$$

where the parameter σ is the kernel width.

In a standard logistic regression, the conditional distribution for t is given by

$$p\left(t|\boldsymbol{\beta}\right) = \sigma\left(\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x})\right)^t \left\{1 - \sigma\left(\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x})\right)\right\}^{1-t}$$

where $\sigma(u)$ is the logistic function defined as $\sigma(u) = 1/\left(1 + e^{-u}\right)$. Assuming independent and identically distributed data, the likelihood function can be written as

$$p\left(\mathbf{t}|\boldsymbol{\beta}\right) = \prod_{i=1}^{N} p\left(t_i|\boldsymbol{\beta}\right) = \prod_{i=1}^{N} \sigma\left(\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i)\right)^{t_i} \left\{1 - \sigma\left(\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i)\right)\right\}^{1-t_i}.$$

The model coefficients $\boldsymbol{\beta}$ can be estimated by the maximum likelihood approach which can be formulated as the following optimization problem in the loss function framework:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} l\left\{(2t_i - 1) f\left(\boldsymbol{\phi}(\mathbf{x}_i)\right)\right\} \tag{1}$$

where $l(u) = \ln\left(1 + e^{-u}\right)$ denotes the logistic loss function. It should be noted that the solution of minimizing the sum of loss functions is equivalent to that of maximizing the log likelihood function, that is

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} l\left\{(2t_i - 1) f\left(\boldsymbol{\phi}(\mathbf{x}_i)\right)\right\} \Leftrightarrow \max_{\boldsymbol{\beta}} \ln p\left(\mathbf{t}|\boldsymbol{\beta}\right) \Leftrightarrow \max_{\boldsymbol{\beta}} \sum_{i=1}^{N} \ln p\left(t_i|\boldsymbol{\beta}\right). \tag{2}$$

To obtain a robust classification result, a weighting strategy can be employed to the standard logistic regression model in the loss function framework as follows

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} w_i l\left\{(2t_i - 1) f\left(\boldsymbol{\phi}(\mathbf{x}_i)\right)\right\}$$

where w_i is a weight associated with the i-th observation. If a small weight is given to an outlying observation, the effect of an outlier can be reduced and therefore a robust decision boundary can be obtained. Then, one question is raised: how the concept of a weighted loss can be transformed into the maximum likelihood approach. From (2), the following relationship can be obtained:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} w_i l \left\{(2t_i - 1)\, f\left(\boldsymbol{\phi}(\mathbf{x}_i)\right)\right\} \Leftrightarrow \max_{\boldsymbol{\beta}} \sum_{i=1}^{N} w_i \ln p\left(t_i|\boldsymbol{\beta}\right) \Leftrightarrow \max_{\boldsymbol{\beta}} \sum_{i=1}^{N} \ln p\left(t_i|\boldsymbol{\beta}\right)^{w_i}. \quad (3)$$

Therefore, the concept of a weighted loss can be dealt with in the maximum likelihood approach by replacing $p\left(t_i|\boldsymbol{\beta}\right)$ with $p\left(t_i|\boldsymbol{\beta}\right)^{w_i}$.

To avoid the overfitting problem while considering a complex model, the regularization concept has been used in machine learning. By employing the regularization concept to the original logistic regression, the formulation in Eq. (1) can be extended as follows

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} l \left\{(2t_i - 1)\, f\left(\boldsymbol{\phi}(\mathbf{x}_i)\right)\right\} + \lambda J(f)$$

where $\lambda > 0$ is a regularization parameter which controls the smoothness of a decision boundary and $J(f)$ denotes a regularization term which represents a penalty for a complex decision boundary. Note that the penalized logistic regression uses the L_2 penalty: $J(f) = \boldsymbol{\beta}^T\boldsymbol{\beta}$ (Lin et al. 2000; Park and Liu 2011), and therefore it does not have a sparse property. While most of classification algorithms achieve the sparse solution by adding the regularization term to the sum of the loss function explicitly, the Bayesian approach such as the RVM apply the regularization through a specific prior distribution over model coefficients. Consequently, both the penalized logistic regression and RVM for classification can be regarded as the standard logistic regression with the regularization on model coefficients, while the RVM has an advantage for obtaining a sparse solution.

3 Robust RVM for classification with variational inference

In classification, it is not possible to directly seek the posterior distributions over the model coefficients since the logistic likelihood function is not suitable to be combined with a Gaussian prior. To resolve this issue, Jaakkola and Jordan (2000) proposed a transformed logistic function which depends on the model coefficients at most quadratically in the exponent, and analyzed the logistic regression model with a Gaussian prior over the model coefficients in the Bayesian framework using the transformed logistic function. Based on their study, Bishop and Tipping (2000) introduced an alternative training algorithm in the framework of variational inference for the RVM.

3.1 Transformation of the logistic function

A lower bound on the logistic function which has the functional form of a Gaussian can be obtained as follows (Jaakkola and Jordan 2000). First, the log of the logistic function $\sigma(u)$ is decomposed as follows:

$$\ln \sigma(u) = -\ln(1 + e^{-u}) = \frac{u}{2} - \ln\left(e^{u/2} + e^{-u/2}\right). \quad (4)$$

Note that the function $f(u) = -\ln\left(e^{u/2} + e^{-u/2}\right)$ is a convex function with respect to the variable u^2. Since a tangent surface to a convex function is a global lower bound for the function, the global lower bound on $f(u)$ can be obtained with a first order Taylor expansion in the variable u^2 at the point ξ (called a variational parameter in the variational inference framework). That is,

$$f(u) \geq f(\xi) + \frac{\partial f(\xi)}{\partial (\xi^2)} \left(u^2 - \xi^2\right) = -\frac{\xi}{2} + \ln \sigma(\xi) + \frac{1}{4\xi} \tanh\left(\frac{\xi}{2}\right) \left(u^2 - \xi^2\right).$$

Combining this lower bound on $f(u)$ with Eq. (4), the lower bound on the logistic function can be obtained as

$$\sigma(u) \geq \sigma(\xi) \exp\left\{\frac{u-\xi}{2} - \lambda(\xi)\left(u^2 - \xi^2\right)\right\} \tag{5}$$

where $\lambda(\xi) = \frac{1}{4\xi} \tanh\left(\frac{\xi}{2}\right) = \frac{1}{2\xi}\left\{\sigma(\xi) - \frac{1}{2}\right\}$. The bound has the form of the exponential-quadratic function of u, which makes the Bayesian approach analytically tractable.

Again, the conditional distribution for t_i can be written as

$$\begin{aligned} p(t_i|\boldsymbol{\beta}) &= \sigma\left(\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i)\right)^{t_i} \left\{1 - \sigma\left(\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i)\right)\right\}^{1-t_i} \\ &= \left(\frac{1}{1+e^{-\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i)}}\right)^{t_i} \left(1 - \frac{1}{1+e^{-\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i)}}\right)^{1-t_i} \\ &= e^{\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i) t_i} \sigma\left(-\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i)\right). \end{aligned}$$

Then, the following relationship holds due to (5):

$$\begin{aligned} p(t_i|\boldsymbol{\beta}) &= e^{\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i) t_i} \sigma\left(-\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i)\right) \\ &\geq \sigma(\xi_i) \exp\left\{\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i) t_i - \frac{\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i) + \xi_i}{2} - \lambda(\xi_i)\left(\left(\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i)\right)^2 - \xi_i^2\right)\right\} \\ &\equiv h(\boldsymbol{\beta}, \xi_i). \end{aligned}$$

Therefore, the likelihood function can be written as

$$p(\mathbf{t}|\boldsymbol{\beta}) = \prod_{i=1}^{N} p(t_i|\boldsymbol{\beta}) \geq \prod_{i=1}^{N} h(\boldsymbol{\beta}, \xi_i).$$

Consequently, from (3), the modified likelihood function to downweight outliers is given by

$$\begin{aligned} p(\mathbf{t}|\boldsymbol{\beta}, \mathbf{w}) &= \prod_{i=1}^{N} p(t_i|\boldsymbol{\beta})^{w_i} = \prod_{i=1}^{N} \left[\sigma\left(\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i)\right)^{t_i} \left\{1 - \sigma\left(\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i)\right)\right\}^{1-t_i}\right]^{w_i} \\ &\geq \prod_{i=1}^{N} \left[\sigma(\xi_i) \exp\left\{\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i) t_i - \frac{\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i) + \xi_i}{2} - \lambda(\xi_i)\left(\left(\boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}_i)\right)^2 - \xi_i^2\right)\right\}\right]^{w_i} \\ &= \prod_{i=1}^{N} h(\boldsymbol{\beta}, \xi_i, w_i) \equiv h(\boldsymbol{\beta}, \boldsymbol{\xi}, \mathbf{w}). \end{aligned}$$

3.2 Derivation of variational inference for robust RVM

Using the results of Sect. 3.1, a robust RVM for classification can be analyzed in the fully Bayesian framework similar to Bishop and Tipping (2000). First, the following prior distributions over the model parameters are assumed.

$$p\left(\boldsymbol{\beta}|\boldsymbol{\alpha}\right)=\prod_{i=0}^{N}N\left(\beta_i|0,\alpha_i^{-1}\right)$$

$$p\left(\boldsymbol{\alpha}|a,b\right)=\prod_{i=0}^{N}Gamma\left(\alpha_i|a,b\right)$$

$$p\left(\mathbf{w}|c,d\right)=\prod_{i=1}^{N}Gamma\left(w_i|c,d\right)$$

The Bayesian posterior distribution over all unknowns can be written as follows by applying the Bayes' rule:

$$p\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\,|\mathbf{t},a,b,c,d\,\right)=\frac{p\left(\mathbf{t}|\boldsymbol{\beta},\mathbf{w}\right)p\left(\boldsymbol{\beta}|\boldsymbol{\alpha}\right)p\left(\boldsymbol{\alpha}|a,b\right)p\left(\mathbf{w}|c,d\right)}{p\left(\mathbf{t}\right)}. \tag{6}$$

It is not feasible to evaluate the posterior distribution based on Eq. (6) since its denominator contains an intractable integration. Therefore, the posterior distribution is approximated by using the derived variational inference method in this study. For the variational inference, $\ln p(\mathbf{t})$ is expressed as the difference of two terms:

$$\ln p(\mathbf{t})=\ln p\left(\mathbf{t},\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)-\log p\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}|\mathbf{t}\right).$$

If an arbitrary distribution $Q(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w})$ which is an approximating distribution of $p(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}|\mathbf{t})$ is introduced, then $\ln p(\mathbf{t})$ can be written as

$$\begin{aligned}\ln p(\mathbf{t})&=\ln\left\{\frac{p\left(\mathbf{t},\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)}{Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)}\right\}-\ln\left\{\frac{p\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}|\mathbf{t}\right)}{Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)}\right\}\\&=\mathcal{L}\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\right]+KL\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)||p\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}|\mathbf{t}\right)\right]\end{aligned}$$

where

$$\mathcal{L}\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\right]=\iiint Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\ln\left\{\frac{p\left(\mathbf{t},\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)}{Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)}\right\}d\boldsymbol{\beta}d\boldsymbol{\alpha}d\mathbf{w}$$

$$KL\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)||p\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}|\mathbf{t}\right)\right]=-\iiint Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\ln\left\{\frac{p\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}|\mathbf{t}\right)}{Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)}\right\}d\boldsymbol{\beta}d\boldsymbol{\alpha}d\mathbf{w}.$$

$KL\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)||p\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}|\mathbf{t}\right)\right]$ is the Kullback–Leibler divergence between $Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)$ and the posterior distribution $p\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}|\mathbf{t}\right)$. Since the Kullback–Leibler divergence is nonnegative, with equality if, and only if, $Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)=p\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}|\mathbf{t}\right)$, $\ln p\left(\mathbf{t}\right)$ is always greater or equal to $\mathcal{L}\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\right]$. In other words $\mathcal{L}\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\right]$ is a lower bound on $\ln p(\mathbf{t})$. Therefore, since maximizing the lower bound $\mathcal{L}\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\right]$ with respect to $Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)$is equivalent to minimizing $KL\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)||p\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}|\mathbf{t}\right)\right]$, the approximating distribution $Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)$ of the posterior $p\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}|\mathbf{t}\right)$ can be obtained.

However, it is not possible to obtain the variational posterior distributions by directly maximizing the lower bound $\mathcal{L}\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\right]$. Based on the results of Sect. 3.1, the lower bound $\mathcal{L}\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\right]$ is replaced with a further bound $\tilde{\mathcal{L}}\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\right]$ using the inequality

$$\mathcal{L}\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\right]\geq\iiint Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\ln\left\{\frac{\tilde{p}\left(\mathbf{t},\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)}{Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)}\right\}d\boldsymbol{\beta}d\boldsymbol{\alpha}d\mathbf{w}\equiv\tilde{\mathcal{L}}\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\right]$$

where $\tilde{p}\left(\mathbf{t},\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)=h\left(\boldsymbol{\beta},\mathbf{w},\boldsymbol{\xi}\right)p\left(\boldsymbol{\beta}|\boldsymbol{\alpha}\right)p\left(\boldsymbol{\alpha}|a,b\right)p\left(\mathbf{w}|c,d\right)$. The inequality always holds since $p\left(\mathbf{t},\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\geq\tilde{p}\left(\mathbf{t},\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)$ [refer to Property 3.2 in Ma and Leijon (2011)]. Assuming $\boldsymbol{\beta}$, $\boldsymbol{\alpha}$ and $\mathbf{w}$ are separable, i.e., $Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)=Q_{\boldsymbol{\beta}}\left(\boldsymbol{\beta}\right)Q_{\boldsymbol{\alpha}}\left(\boldsymbol{\alpha}\right)Q_{\mathbf{w}}\left(\mathbf{w}\right)$, the new lower bound $\tilde{\mathcal{L}}\left[Q\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\right]$ is maximized using the following results:

$$\begin{aligned}\ln Q_{\boldsymbol{\beta}}\left(\boldsymbol{\beta}\right) &= \mathbb{E}_{\boldsymbol{\alpha},\mathbf{w}}\left[\ln \tilde{p}\left(\mathbf{t},\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\right] + \text{constant}\\ \ln Q_{\boldsymbol{\alpha}}\left(\boldsymbol{\alpha}\right) &= \mathbb{E}_{\boldsymbol{\beta},\mathbf{w}}\left[\ln \tilde{p}\left(\mathbf{t},\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\right] + \text{constant}\\ \ln Q_{\mathbf{w}}\left(\mathbf{w}\right) &= \mathbb{E}_{\boldsymbol{\beta},\boldsymbol{\alpha}}\left[\ln \tilde{p}\left(\mathbf{t},\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right)\right] + \text{constant}\end{aligned}$$

where $\ln \tilde{p}\left(\mathbf{t},\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w}\right) = \ln h\left(\boldsymbol{\beta},\mathbf{w},\boldsymbol{\xi}\right) + \ln p\left(\boldsymbol{\beta}|\boldsymbol{\alpha}\right) + \ln p\left(\boldsymbol{\alpha}|a,b\right) + \ln p\left(\mathbf{w}|c,d\right)$.

Finally, the variational posterior distributions can be evaluated analytically as follows

$$Q_{\boldsymbol{\beta}}\left(\boldsymbol{\beta}\right) = N\left(\boldsymbol{\beta}|\boldsymbol{\mu},\boldsymbol{\Sigma}\right) \tag{7}$$

$$Q_{\boldsymbol{\alpha}}\left(\boldsymbol{\alpha}\right) = \prod_{i=0}^{N} Gamma\left(\alpha_i|\tilde{a},\tilde{b}_i\right) \tag{8}$$

$$Q_{\mathbf{w}}\left(\mathbf{w}\right) = \prod_{i=1}^{N} Gamma\left(w_i|\tilde{c},\tilde{d}_i\right) \tag{9}$$

where

$$\boldsymbol{\Sigma} = \left\{\mathbf{A} + 2\sum_{i=1}^{N}\mathbb{E}\left(w_i\right)\lambda\left(\xi_i\right)\boldsymbol{\phi}\left(\mathbf{x}_i\right)\boldsymbol{\phi}\left(\mathbf{x}_i\right)^T\right\}^{-1} \tag{10}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}\left\{\sum_{i=1}^{N}\mathbb{E}\left(w_i\right)\left(t_i - \frac{1}{2}\right)\boldsymbol{\phi}\left(\mathbf{x}_i\right)\right\} \tag{11}$$

$$\tilde{a} = a + \frac{1}{2} \tag{12}$$

$$\tilde{b}_i = b + \frac{\mathbb{E}\left(\beta_i^2\right)}{2} \tag{13}$$

$$\tilde{c} = c \tag{14}$$

$$\begin{aligned}\tilde{d}_i &= d - \left[\ln\sigma\left(\xi_i\right) + \left(t_i - \frac{1}{2}\right)\mathbb{E}\left(\boldsymbol{\beta}\right)^T\boldsymbol{\phi}\left(\mathbf{x}_i\right) - \frac{\xi_i}{2} - \lambda\left(\xi_i\right)\left\{\boldsymbol{\phi}\left(\mathbf{x}_i\right)^T\mathbb{E}\left(\boldsymbol{\beta}\boldsymbol{\beta}^T\right)\boldsymbol{\phi}\left(\mathbf{x}_i\right) - \xi_i^2\right\}\right]\\ &= d - \ln\mathbb{E}\left(h\left(\boldsymbol{\beta},\xi_i\right)\right).\end{aligned} \tag{15}$$

Here, $\mathbf{A}$ in Eq. (10) denotes a diagonal matrix with $\mathbb{E}(\alpha_i)$ as the i-th diagonal element. Note that the resulting posterior distributions over parameters in (7)–(9) have the same parametric form as the prior distributions. Each posterior distribution is governed by some hyperparameters, for example $Q_{\boldsymbol{\beta}}(\boldsymbol{\beta})$ is characterized by computing the posterior mean vector $\boldsymbol{\mu}$ and posterior covariance matrix $\boldsymbol{\Sigma}$.

The variational parameters $\{\xi_i\}$ also need to be optimized. Arranging terms that are dependent of $\boldsymbol{\xi}$, we have

$$\tilde{\mathcal{L}}[Q(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w})] = \iint Q_{\boldsymbol{\beta}}(\boldsymbol{\beta})Q_{\mathbf{w}}(\mathbf{w})\ln h\left(\boldsymbol{\beta},\mathbf{w},\boldsymbol{\xi}\right)d\boldsymbol{\beta}d\mathbf{w} + \text{constant}.$$

The re-estimation equation for parameter ξ_i can be obtained by directly maximizing the bound $\tilde{\mathcal{L}}[Q(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{w})]$ with respect to ξ_i, leading to the following result:

$$(\xi_i)^2 = \boldsymbol{\phi}(\mathbf{x}_i)^T\mathbb{E}\left(\boldsymbol{\beta}\boldsymbol{\beta}^T\right)\boldsymbol{\phi}\left(\mathbf{x}_i\right) \tag{16}$$

The weight strategy with the variational inference in our proposed procedure can be interpreted in the framework of the loss function. From a loss function point of view, a bounded loss function is desired to reduce the effect of outliers. The bounded loss function implies that the loss value for an outlier located further away from the decision boundary is limited to some upper bound so that the outlier cannot further influence the decision boundary. Proposition 1 shows that the weights with a Gamma prior in our model make the logistic loss function $l(u)$ bounded (see Appendix 1 for the proof).

Proposition 1 *The weight value* $\mathbb{E}(w)$ *obtained from the variational posterior distribution* $Q_{\mathbf{w}}(\mathbf{w})$ *makes the logistic loss function* $l(u)$ *bounded.*

From the results in (7)–(9), it is observed that the variational posterior distributions are coupled, for example $Q_{\boldsymbol{\beta}}(\boldsymbol{\beta})$ depends on the expectation values of $Q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})$ and vice versa. With some initial values for hyperparameters a, b, c, d, and variational parameters $\{\xi_i\}$, the variational posterior distributions can be computed by iteratively re-estimating the posterior distributions until a convergence criterion is satisfied. In this paper, the hyperparameters a and b of the prior distribution of $\boldsymbol{\alpha}$ are set to small values (e.g., $a = b = 10^{-5}$) in order to make the prior non-informative. For the hyperparameters c and d of the prior distribution of weight $\mathbf{w}$, it is natural to think that the weight value $\mathbb{E}(w)$ lies within the interval [0, 1]. Proposition 2 shows that how the hyperparameters c and d should be determined for $\mathbb{E}(w)$ to satisfy the interval condition (see Appendix 2 for the proof).

Proposition 2 *The weight value* $\mathbb{E}(w)$ *always lies within the interval [0, 1], if and only if, the hyperparameters c and d are equal to each other, i.e.* $0 \leq \mathbb{E}(w) \leq 1 \Leftrightarrow c = d \equiv r$.

Note that $\ln \mathbb{E}(h(\boldsymbol{\beta}, \xi))$ is an approximation of the logistic loss function $l\left\{(2t-1)\hat{\boldsymbol{\beta}}^T \boldsymbol{\phi}(\mathbf{x})\right\}$. By Proposition 2, the weight value $\mathbb{E}(w)$ can be approximated as

$$\mathbb{E}(w) = \frac{r}{r - \ln \mathbb{E}(h(\boldsymbol{\beta}, \xi))} \approx \frac{r}{r + l\left\{(2t-1)\hat{\boldsymbol{\beta}}^T \boldsymbol{\phi}(\mathbf{x})\right\}}.$$

Then the weighted logistic loss function is given by

$$\mathbb{E}(w) l\left\{(2t-1)\hat{\boldsymbol{\beta}}^T \boldsymbol{\phi}(\mathbf{x})\right\} \approx \frac{r}{r + l\left\{(2t-1)\hat{\boldsymbol{\beta}}^T \boldsymbol{\phi}(\mathbf{x})\right\}} l\left\{(2t-1)\hat{\boldsymbol{\beta}}^T \boldsymbol{\phi}(\mathbf{x})\right\}$$

The hyperparameter r is determined to maximize the rate of change of loss around the decision boundary. This approach is reasonable in the sense that observations near the decision boundary are more important than those far from the decision boundary. Based on such criterion, the hyperparameter r is set to $r = 2 - \ln 2$ in our proposed model by solving the following equation

$$\frac{\partial}{\partial u}\left[\frac{r}{r + l(u)} l(u)\right]\Bigg|_{u=0} = 0.$$

The weighted logistic loss function with $r = 2 - \ln 2$ is plotted in Fig. 2 together with the original logistic loss function. From this figure, it is observed that the weighted logistic loss function assigns an approximately consistent loss value to an observation located far away from the decision boundary, and therefore its influence on the decision boundary is effectively bounded. From Proposition 1, the upper bound of the weighted logistic function is $r = 2 - \ln 2$.

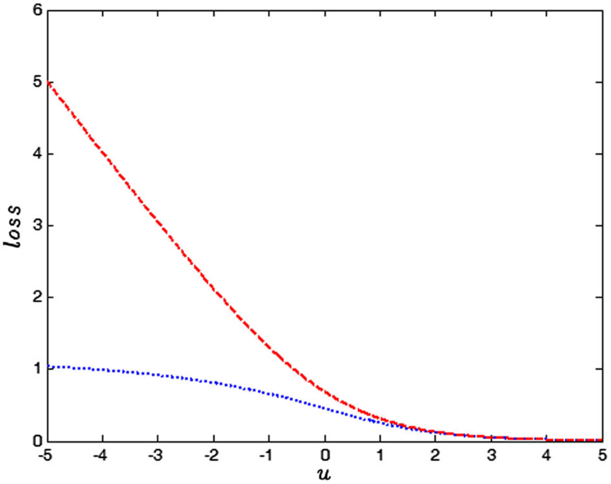


Fig. 2 Plots of the weighted logistic loss function with $r = 2 - \ln 2$ (*dotted line*) and the original logistic loss function (*dashed line*)

3.3 Variational lower bound and model comparison

The lower bound $\tilde{\mathcal{L}}\left[Q\left(\boldsymbol{\beta}, \boldsymbol{\alpha}, \mathbf{w}\right)\right]$ can be evaluated as follows:

$$\begin{aligned}\tilde{\mathcal{L}}\left[Q\left(\boldsymbol{\beta}, \boldsymbol{\alpha}, \mathbf{w}\right)\right] = {} & \mathbb{E}\left[\ln h\left(\boldsymbol{\beta}, \mathbf{w}, \boldsymbol{\xi}\right)\right] + \mathbb{E}\left[\ln p\left(\boldsymbol{\beta}|\boldsymbol{\alpha}\right)\right] + \mathbb{E}\left[\ln p\left(\boldsymbol{\alpha}|a, b\right)\right] \\ & + \mathbb{E}\left[\ln p\left(\mathbf{w}|c, d\right)\right] - \mathbb{E}\left[\ln Q_{\boldsymbol{\beta}}\left(\boldsymbol{\beta}\right)\right] - \mathbb{E}\left[\ln Q_{\boldsymbol{\alpha}}\left(\boldsymbol{\alpha}\right)\right] - \mathbb{E}\left[\ln Q_{\mathbf{w}}\left(\mathbf{w}\right)\right]. \end{aligned} \tag{17}$$

The lower bound $\tilde{\mathcal{L}}\left[Q\left(\boldsymbol{\beta}, \boldsymbol{\alpha}, \mathbf{w}\right)\right]$ is an important quantity in the variational inference and increases at each iteration. Thus, it can be used to check the convergence of the iterative procedure. Moreover, we can select the best model by comparing the values of lower bound of the candidate models. This means that models can be compared directly on the training data. This is a very valuable property since it allows all available data to be used for training and avoids multiple estimation procedures for each model with a common validation method such as cross validation.

3.4 Iterative algorithm

Update Eqs. (10)–(15) given the hyperparameters. The training procedure of the proposed method can be summarized as follows:

Step 1 Choose initial values for a, b, c, d and $\{\xi_i\}$ according to the guideline in Sect. 3.3.
Step 2 Update the hyperparameters $\boldsymbol{\Sigma}, \boldsymbol{\mu}, \tilde{a}, \tilde{b}_i, \tilde{c}$ and $\tilde{d}_i$ using Eqs. (10)–(15). The variational posterior distributions in (7)–(9) can be characterized using these updated hyperparameter values.
Step 3 Re-estimate the variational parameters $\{\xi_i\}$ using Eq. (16).
Step 4 Compute the lower bound value $\tilde{\mathcal{L}}\left[Q\left(\boldsymbol{\beta}, \boldsymbol{\alpha}, \mathbf{w}\right)\right]$ in (17) using the posterior distributions obtained from Step 2.
Step 5 Repeat Steps 2–4 until convergence. Specifically, iterate until the change in the lower bound value $\tilde{\mathcal{L}}\left[Q\left(\boldsymbol{\beta}, \boldsymbol{\alpha}, \mathbf{w}\right)\right]$ is smaller than a predefined threshold, e.g. 10^{-5}.

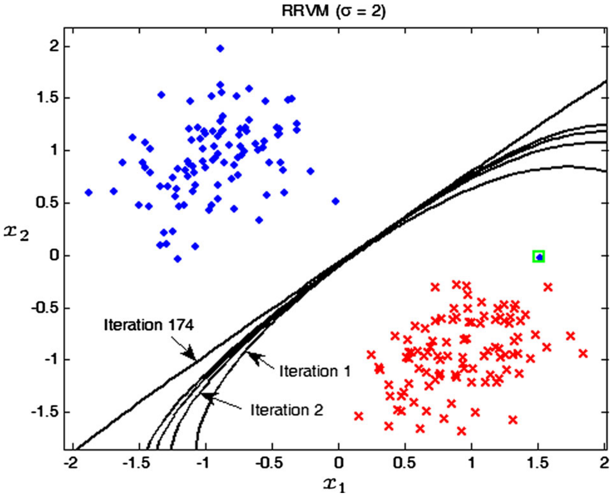


Fig. 3 Illustration of the change of a decision boundary at each iteration (The outlier is shown in a *square*)

Figure 3 shows the results of applying the proposed iterative algorithm to the simulated example in Fig. 1 (with the RBF kernel). We can observe that the effect of the outlier is iteratively reduced by assigning a smaller weight to it. After convergence, a decision boundary which is robust to the outlier is obtained.

4 Computational experiments

In this section, computational experiments are conducted using two synthetic and three real data sets to verify the robustness of the proposed method (RRVM) compared to other classification algorithms: one-nearest neighbor (1-NN), k-nearest neighbor (k-NN), SVM and RVM.

4.1 Experimental setup and performance measures

For k-NN, the number of nearest neighbor k should be selected. In this experiment, a five-fold cross validation procedure is applied to the training data set, and then the optimal number of k giving the minimum error rate is selected. The similar procedure is applied to optimize model parameters of the SVM and RVM. The SVM has two model parameters: the regularization parameter C and the kernel parameter (e.g., in the case of the RBF kernel, the width σ of the kernel function) while the RVM has a single parameter (the kernel parameter value). The proposed RRVM also has the kernel parameter as a single model parameter. While the parameters of the SVM should be optimized through the cross validation procedure which is computationally demanding, the parameter of the RVM can be selected efficiently by comparing the lower bound values.

The generalization performance of each method is evaluated in terms of three performance measures: the error rate (ERR), area under the curve of receiver operating characteristics curve (AUC) and root mean squared error (RMSE), which represent a threshold measure, rank measure and probability measure, respectively (Caruana and Niculescu-Mizil 2004). The

ERR has been widely used as the main criterion for comparing the generalization performance of classifiers. It is defined as the proportion of misclassified observations relative to the total number of observations. However, it does not consider how close a predicted value is to a threshold, in other words it considers only whether the predicted value is above or below a threshold. As a result, probability outcomes of classifiers are completely ignored. The rank measure such as AUC (Ling et al. 2003) deals with the predicted value differently from the threshold metric. If the observations in a test set are ordered by their predicted values, the rank measure evaluates how well the ordering ranks positive classes above negative classes. Namely, the rank measure can be viewed as a summary of the performance of a classifier across all possible thresholds. The RMSE depends on the predicted values, not on how the values fall relative to a threshold or relative to each other. It measures how much predictions deviated from the true target values (Fang and Jeong 2008). Note that smaller values of the ERR and RMSE mean the better classification ability of the model, while for the AUC, higher is better.

4.2 Experimental results

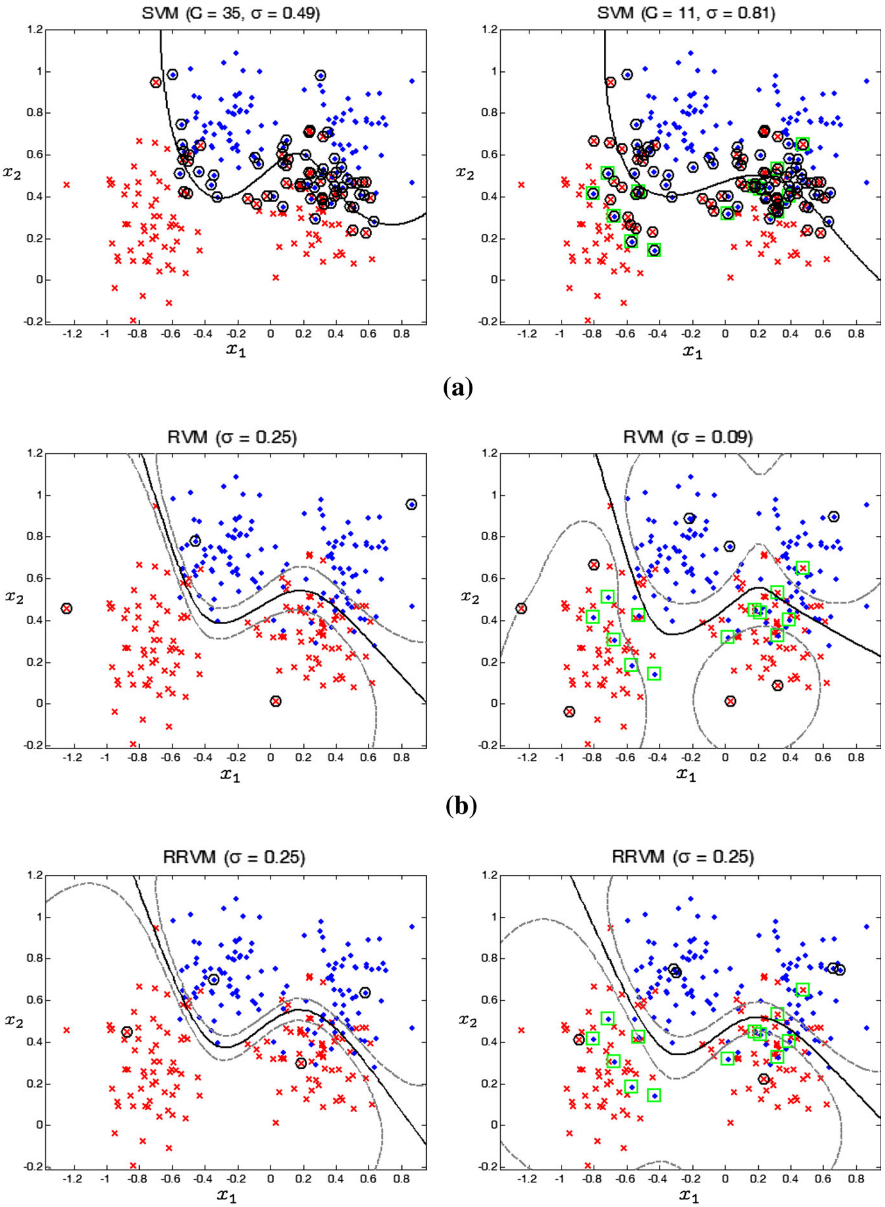
4.2.1 Synthetic data set

Ripley's synthetic data set[1] and Banana data set (Ratsch et al. 2001) are used to verify the robustness of the RRVM compared to other classifiers. Since these data sets are two dimensional, the effect of outliers on the decision boundary can be illustrated graphically. Figures 4 and 5 illustrate the effect of outliers on the decision boundaries obtained from the SVM, RVM and the RRVM for Ripley's data set and Banana data set, respectively. The dashed lines in (b) and (c) of each figure correspond to output probabilities of 0.25 and 0.75. Note that the SVM does not provide such probabilistic information. From the figures, it can be observed that the SVM and RVM are not robust to the outliers, i.e. the decision boundaries are distorted by a few outliers. In contrast to them, the RRVM is more insensitive to outliers since it reduces the effect of outliers by giving a small weight to them. In terms of the sparsity, the RRVM preserves the sparsity, i.e. the number of non-zero coefficient is small enough, although the training data set contains outliers. For the Ripley's data set contaminated by outliers, only 6 observations are chosen as *relevance vectors* of the RRVM (see Fig. 4c). It is should be noted that the set of *support vectors* of the SVM includes outliers, i.e. outliers are explicitly shown in the final model since their coefficients are nonzero.

To evaluate the generalization performance in terms of the robustness, each data set is randomly divided into the training (60 %) and test data sets (40 %). Then, the class labels of the training data set are contaminated by randomly choosing *perc* of their observations and changing the corresponding class labels to the other class, where *perc* = 0, 5 and 10 % (Wu and Liu 2007; Park and Liu 2011). For each value of *perc*, the experiment is repeated 50 times.

Table 1 shows the summaries of classification results over 50 repetitions for Ripley's data set with *perc* = 0, 5 and 10 %, respectively. Standard deviations are in parenthesis. From Table 1, it is clearly shown that the generalization performances of the RRVM are consistently better than other methods even if the training data set is contaminated by the outliers. In addition, in terms of the sparsity, the final models from the RVM and RRVM have much smaller non-zero coefficients than that from the SVM as can be seen in Table 2.

[1] It can be found at http://www.stats.ox.ac.uk/pub/PRNN/.



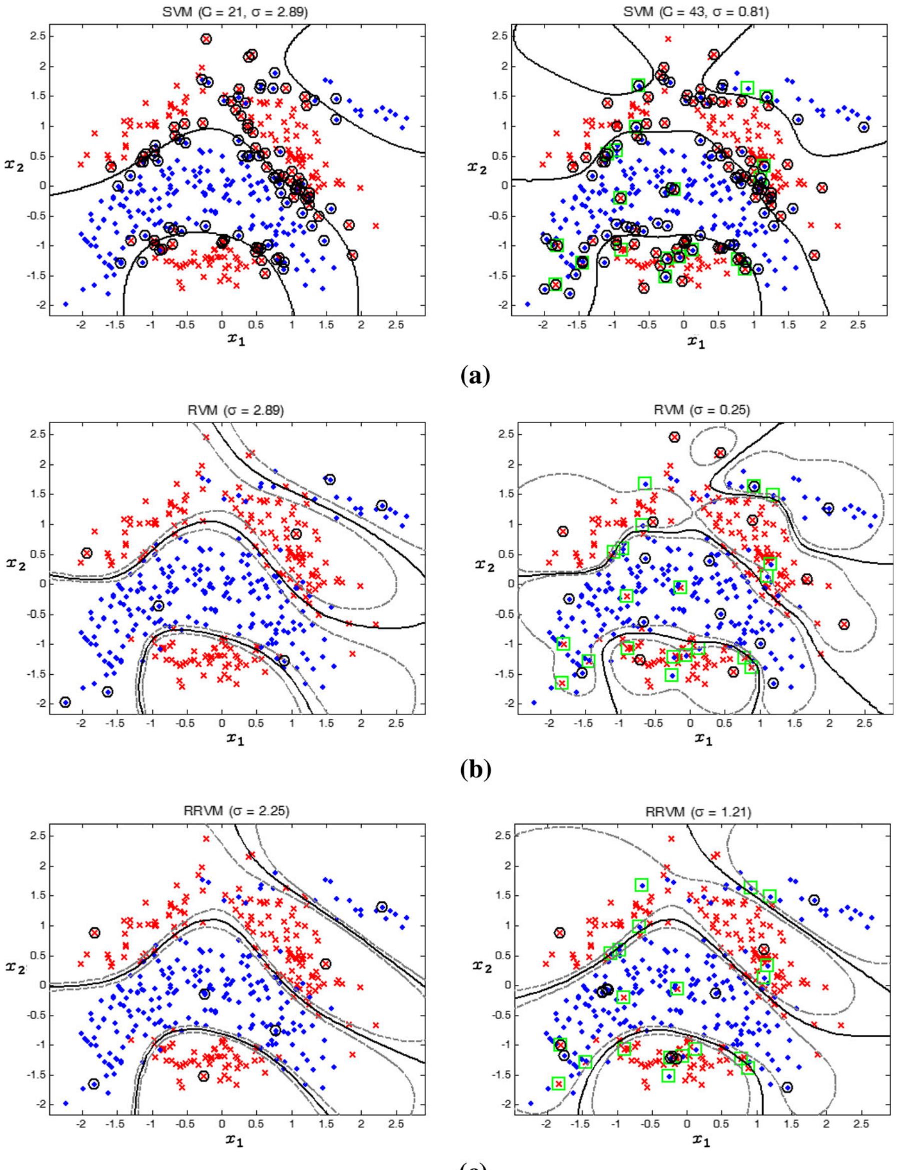
(a)

(b)

(c)

Fig. 4 Illustration of decision boundaries of **a** SVM, **b** RVM, and **c** RRVM for Ripley's data set. The *circled* observations represent the support vectors or the relevance vectors. The simulated outliers are shown in *squares*. **a** Decision boundary from SVM (*left* without outliers, *right* with outliers). **b** Decision boundary from RVM (*left* without outliers, *right* with outliers). **c** Decision boundary from RRVM (*left* without outliers, *right* with outliers)

The computation time of each method for model selection and training is summarized in Table 3. From the table, the RRVM takes relatively short time for model selection compared to the SVM and RVM since it does not need a grid search such as cross validation to select the



(a)

(b)

(c)

Fig. 5 Illustration of decision boundaries of **a** SVM, **b** RVM, and **c** RRVM for Banana data set. The *circled* observations represent the support vectors or the relevance vectors. The simulated outliers are shown in *squares*. **a** Decision boundary from SVM (*left* without outliers, *right* with outliers). **b** Decision boundary from RVM (*left* without outliers, *right* with outliers). **c** Decision boundary from RRVM (*left* without outliers, *right* with outliers)

optimal model parameters. However, the RRVM takes relatively long time for model training since its iterative algorithm contains a matrix inverse computation which has an $O(N^3)$ time complexity inherited from Bishop and Tipping (2000). The computational environment is Windows 7 with Intel Core i5-750 2.66 GHz and 4GB RAM. The SVM is implemented using

Table 1 Generalization performance of classification methods for Ripley's data set

Perc	Classifier	ERR	AUC	RMSE
0	1-NN	13.24 (1.15)	86.76 (1.15)	0.3636 (0.0158)
	k-NN	9.73 (1.74)	95.32 (3.54)	0.2796 (0.0357)
	SVM	9.85 (0.82)	96.01 (1.04)	0.2700 (0.0124)
	RVM	9.76 (0.61)	96.34 (1.05)	0.2703 (0.0157)
	RRVM	**9.58 (0.72)**	**96.87 (0.48)**	**0.2661 (0.0104)**
5	1-NN	17.98 (1.04)	82.02 (1.04)	0.4239 (0.0122)
	k -NN	11.24 (3.74)	93.10 (5.89)	0.3191 (0.0548)
	SVM	11.87 (3.03)	94.17 (3.02)	0.3068 (0.0311)
	RVM	10.88 (2.70)	95.55 (1.75)	0.2925 (0.0260)
	RRVM	**10.03 (1.31)**	**96.37 (0.93)**	**0.2683 (0.0189)**
10	1-NN	21.62 (1.86)	78.38 (1.86)	0.4646 (0.0200)
	k -NN	15.00 (5.52)	88.84 (8.79)	0.3676 (0.0821)
	SVM	14.91 (5.00)	90.91 (6.01)	0.3414 (0.0485)
	RVM	16.21 (4.60)	91.17 (4.07)	0.3488 (0.0407)
	RRVM	**12.83 (4.48)**	**94.31 (3.05)**	**0.3079 (0.0538)**

The best performance for each *perc* is given in bold

Table 2 The proportion (%) of non-zero coefficients of the estimated models for Ripley's data set

Perc	SVM	RVM	RRVM
0	30.76 (7.76)	2.84 (3.10)	1.88 (0.27)
5	42.32 (6.14)	4.04 (3.73)	2.04 (0.69)
10	50.00 (8.07)	6.80 (4.29)	3.16 (1.85)

LIBSVM software (Chang and Lin 2011), and the source code to run RVM is obtained from Tipping's website.[2] Moreover, the proposed method is developed by MATLAB 7.10.

The experimental results using Banana data set show similar results to those of Ripley's data set. The experiment is repeated 50 times and the results are summarized in Table 4. Also, the number of non-zero coefficients for each model is reported in Table 5. The results show that as the contamination percentage increases, the predictive performances of the classifiers get worse and worse, while the RRVM clearly shows its robustness. In addition, it is shown that the RRVM gives a sparse solution. Furthermore, it is confirmed from Table 6 that the RRVM is competitive with other methods in terms of the computation time since it takes relatively a short time to optimize the model parameters.

4.2.2 Real data

To capture variation among the data sets, the performances on three real data sets (Liver Disorders, Ionosphere, and Breast Cancer Wisconsin[3]) from UCI Machine Learning Repos-

[2] http://www.miketipping.com/.

[3] This breast cancer databases was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg.

Table 3 Computation time of classification methods for Ripley's data set

Time	1-NN	*k*-NN	SVM	RVM	RRVM
Model selection(s)	–	2.85 (0.35)	51.80 (4.65)	36.77 (10.93)	23.01 (3.84)
Model training(s)	0.0473 (0.0280)	0.0425 (0.0029)	0.2515 (0.0331)	0.3364 (0.2398)	0.5443 (0.3564)

Table 4 Generalization performance of classification methods for Banana data set

Perc	Classifier	ERR	AUC	RMSE
0	1-NN	13.73 (0.65)	86.23 (0.62)	0.3705 (0.0087)
	k-NN	11.76 (0.78)	94.88 (1.04)	0.2895 (0.0091)
	SVM	10.96 (0.55)	95.73 (0.57)	0.2830 (0.0072)
	RVM	10.90 (0.55)	**96.11 (0.32)**	**0.2798 (0.0060)**
	RRVM	**10.80 (0.58)**	96.08 (0.26)	0.2856 (0.0068)
5	1-NN	17.31 (1.27)	82.71 (1.31)	0.4158 (0.0154)
	k-NN	14.45 (2.63)	91.18 (5.12)	0.3368 (0.0479)
	SVM	12.13 (1.03)	94.80 (0.74)	0.2997 (0.0094)
	RVM	12.11 (0.95)	94.78 (0.68)	0.3006 (0.0095)
	RRVM	**11.55 (0.66)**	**95.37 (0.39)**	**0.2928 (0.0066)**
10	1-NN	19.55 (1.27)	80.32 (1.21)	0.4419 (0.0142)
	k-NN	16.51 (3.75)	87.08 (7.42)	0.3825 (0.0669)
	SVM	13.05 (2.56)	93.59 (2.54)	0.3140 (0.0262)
	RVM	12.77 (1.72)	93.96 (1.32)	0.3127 (0.0169)
	RRVM	**11.67 (0.93)**	**94.97 (0.91)**	**0.2991 (0.0124)**

The best performance for each *perc* is given in bold

Table 5 The proportion (%) of non-zero coefficients of the estimated models for Banana data set

Perc	SVM	RVM	RRVM
0	29.08 (6.05)	2.63 (0.46)	3.15 (0.66)
5	39.70 (8.36)	3.10 (1.72)	3.58 (0.99)
10	45.08 (5.03)	3.85 (2.30)	4.70 (1.98)

Table 6 Computation time of classification methods for Banana data set

Time	1-NN	*k*-NN	SVM	RVM	RRVM
Model selection(s)	–	2.99 (0.03)	156.27 (7.71)	1039.83 (730.55)	112.60 (6.02)
Model training(s)	0.1435 (0.0218)	0.1371 (0.0066)	1.0527 (0.0185)	1.1722 (0.2958)	2.4391 (0.3560)

itory[4] are investigated (Frank and Asuncion 2010). The Liver Disorders data set has a total of 345 observations with two classes and six input variables which are medical test results for diagnosing liver disorders. The Ionosphere data set contains 351 observations and each observation consists of 34 input variables to distinguish between the "Good" and "Bad" returns of radar. For Breast Cancer Wisconsin (BC Wisconsin) data set, 9 medical attributes are used and 699 observations are collected to diagnose the recurrence of the breast cancer. After deleting 16 observations with missing values, 683 observations are used in this study.

Similar to the experiment for synthetic data sets, in this experiment, each data set is randomly divided into the training (60 %) and test data sets (40 %). Then, the class labels of the training data set are contaminated by randomly choosing $perc = 0$ and 10 % of their observations and changing the corresponding class labels to the other class. Also, for each value of $perc$, the experiment is repeated 50 times.

Table 7 reports the comparative results in terms of the performance measures and computation times for each data set. Although a certain degree of difference between the mean performances is observed, it seems that there is no significant difference in a statistical sense because of large standard deviations. Due to the nature of this experimental setup, it might be happened that predictive performances of all classifiers are degraded simultaneously if randomly selected observations are located very near to the true decision boundary, and these cases cause large standard deviations of performance measures.

To compare the predictive performances of classifiers in a rigorous way, the Friedman and Nemenyi tests were conducted. These tests are well-suited and widely used for the comparison on multiple classifiers and multiple data sets (Demsar 2006). The Friedman test is a nonparametric alternative of the repeated-measures ANOVA. It uses a rank of classifiers for each data set separately, and tests the null hypothesis that all classifiers perform the same. For 3 real data sets contaminated with $perc = 0$ and 10 %, the Friedman tests rejected the null hypothesis regardless of performance measures (p values from the tests can be found in Table 7). The rejection of the null hypothesis means that there exists at least one pair of classifiers with significantly different performances. Therefore, in case of rejection of the null hypothesis, an additional post-hoc test such as the Nemenyi test should follow to identify the significantly different pairs of classifiers. For a detailed description about the statistical tests for the comparison of multiple classifiers including the Friedman and Nemenyi test, please refer to Demsar (2006).

The test results of the case of $perc = 10$ % are depicted in Fig. 6. It shows that regardless of performance measures RRVM performs significantly better than other classifiers if the training data is contaminated by outliers. In terms of the computation time, the proposed method has a longer time than the SVM for BC Wisconsin data set since the data set has more observations compared to other data sets.

To show the robustness of the RRVM in the different point of view, the sensitivity curves of the SVM, RVM, and RRVM are analyzed. The sensitivity curve is a finite sample version of the influence function, which is well-known tool in robustness analysis. It measures the impact of an additional data point on the estimator of interest, and the definition can be found in Definition 2 in Christmann and Steinwart (2004).

Figure 7 shows the sensitivity curves of the SVM, RVM and RRVM from the simulated data in Fig. 1 and the contaminated Ripley data in Fig. 4. These sensitivity curves are obtained by comparing the estimated functions from the training data set including additional outliers with them from the training data set without outliers. The sensitivity values from each classifier are standardized to have a value between -1 and 1 to show the relative difference on the same

[4] http://archive.ics.uci.edu/ml/.

Table 7 Generalization performance and computation time of classification methods for real data sets with $perc = 0$ and 10 %

	Classifiers	Liver disorders		Ionosphere		BC Wisconsin		p value	
		0 %	10 %	0 %	10 %	0 %	10 %	0 %	10 %
ERR	1-NN	37.72 (3.63)	40.28 (3.94)	14.54 (2.04)	25.71 (3.66)	4.58 (1.38)	20.95 (3.34)	2.53×10^{-71}	3.76×10^{-78}
	k-NN	37.75 (3.16)	39.26 (3.49)	11.57 (2.61)	21.46 (5.14)	3.55 (0.86)	7.37 (2.42)		
	SVM	**29.75 (2.74)**	33.93 (3.63)	5.54 (1.70)	9.50 (3.89)	3.26 (0.95)	5.68 (2.51)		
	RVM	32.64 (4.54)	36.36 (6.08)	5.96 (2.49)	9.32 (3.18)	3.77 (0.96)	4.92 (1.52)		
	RRVM	30.76 (3.54)	**31.08 (3.34)**	**5.04 (1.65)**	**7.79 (3.40)**	**3.19 (0.75)**	**3.41 (1.65)**		
AUC	1-NN	61.12 (3.72)	58.94 (3.97)	80.94 (2.85)	69.58 (5.22)	94.57 (1.61)	79.73 (2.38)	1.70×10^{-86}	2.48×10^{-90}
	k-NN	63.61 (3.80)	60.69 (4.04)	86.41 (2.94)	78.83 (7.10)	98.46 (0.74)	85.48 (1.80)		
	SVM	**73.68 (2.88)**	70.66 (3.83)	98.29 (1.14)	95.31 (3.33)	98.98 (0.61)	97.70 (1.88)		
	RVM	69.83 (5.53)	65.84 (7.49)	97.84 (1.77)	95.92 (3.04)	**99.18 (0.51)**	97.94 (1.01)		
	RRVM	73.02 (3.46)	**72.64 (3.69)**	**98.40 (1.12)**	**96.62 (2.14)**	99.17 (0.51)	**99.08 (0.94)**		
100 × RMSE	1-NN	61.34 (3.01)	63.39 (3.13)	38.04 (2.69)	50.59 (3.58)	21.19 (3.13)	45.64 (3.64)	5.62×10^{-86}	4.81×10^{-101}
	k-NN	48.53 (1.83)	51.93 (5.12)	35.55 (2.66)	42.77 (5.20)	17.06 (1.95)	31.70 (4.64)		
	SVM	**45.34 (0.98)**	48.30 (1.84)	20.44 (2.57)	27.56 (4.31)	16.24 (2.28)	22.02 (3.97)		
	RVM	46.94 (2.21)	48.72 (1.58)	21.59 (4.41)	27.05 (3.87)	17.41 (2.38)	20.79 (2.90)		
	RRVM	46.92 (2.46)	**47.22 (1.82)**	**19.98 (2.95)**	**25.58 (4.49)**	**16.20 (2.24)**	**16.33 (3.29)**		
proportion of non-zero coefficients	SVM	69.57 (4.85)	72.90 (7.13)	47.89 (9.60)	63.03 (9.64)	13.51 (3.46)	40.32 (6.56)		
	RVM	5.94 (4.81)	7.42 (7.98)	5.09 (0.46)	7.04 (1.60)	1.39 (0.73)	2.00 (1.42)		
	RRVM	3.60 (0.67)	4.59 (2.36)	2.96 (0.46)	4.19 (1.10)	1.61 (1.35)	2.02 (0.89)		
Model selection time	1-NN	–		–		–			
	k-NN	2.57 (0.02)		2.57 (0.20)		3.19 (0.30)			
	SVM	53.34 (1.75)		53.24 (1.82)		136.93 (12.74)			
	RVM	33.34 (12.33)		368.41 (127.90)		416.08 (147.92)			
	RRVM	29.35 (2.07)		18.01 (3.34)		198.64 (19.80)			

Table 7 continued

	Classifiers	Liver disorders		Ionosphere		BC Wisconsin		*p* value	
		0 %	10 %	0 %	10 %	0 %	10 %	0 %	10 %
Model training time	1-NN	0.0167 (0.0001)		0.0166 (0.0001)		0.0236 (0.0006)			
	k-NN	0.0176 (0.0007)		0.0159 (0.0006)		0.0246 (0.0010)			
	SVM	0.0642 (0.0033)		0.0746 (0.0185)		0.0825 (0.0249)			
	RVM	0.2521 (0.2622)		6.0029 (9.6849)		0.2149 (0.1016)			
	RRVM	1.3902 (0.2622)		1.6825 (0.1539)		4.9613 (1.5824)			

The best performance for each *perc* is given in bold

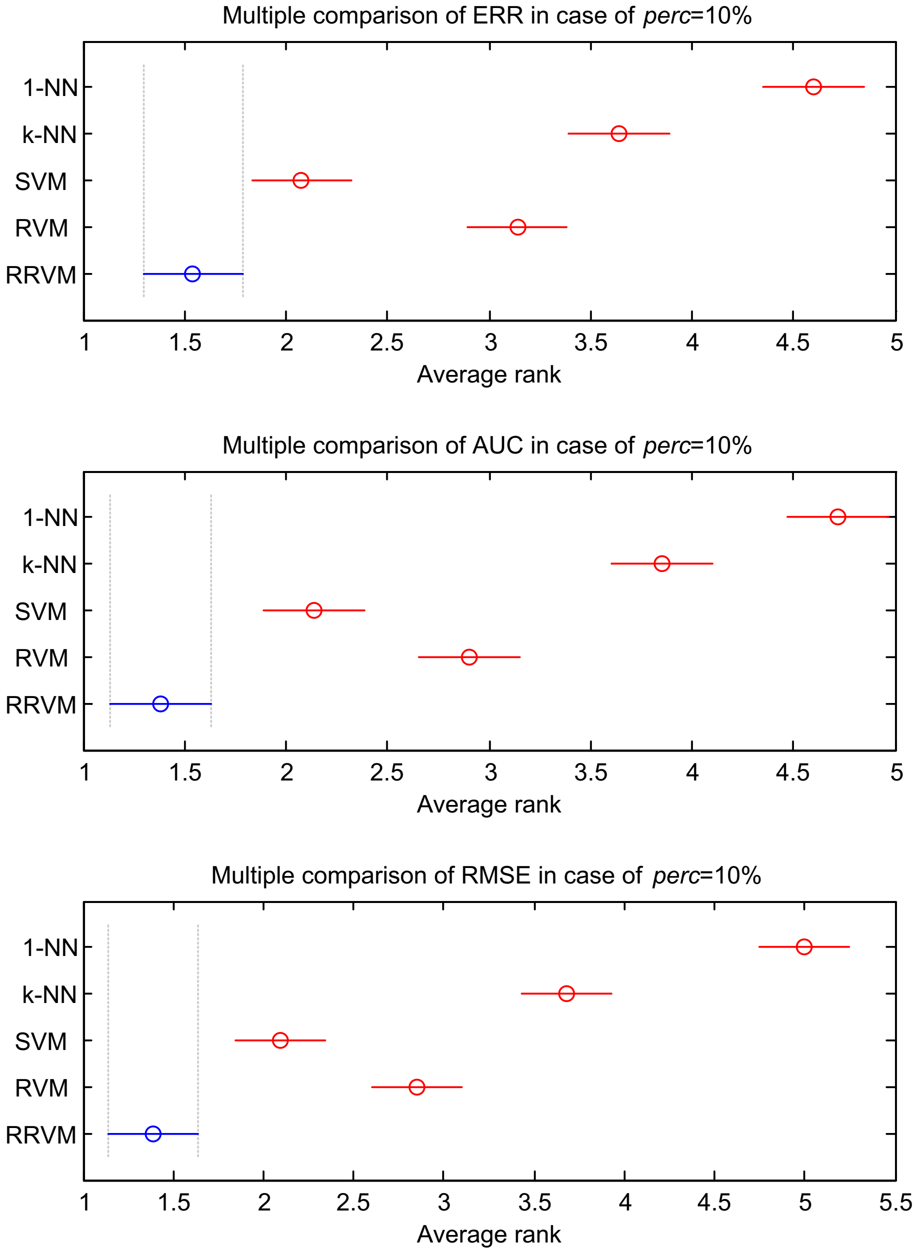


Fig. 6 Multiple comparison results of ERR, AUC, and RMSE in case of *perc* = 10 %. The *circles* and *horizontal lines* represent mean ranks and 95 % confidence intervals, respectively

criteria. From these figures, it can be observed that the sensitivity curve from the RRVM is significantly stable than those from the SVM and RVM, showing the robustness of the RRVM.

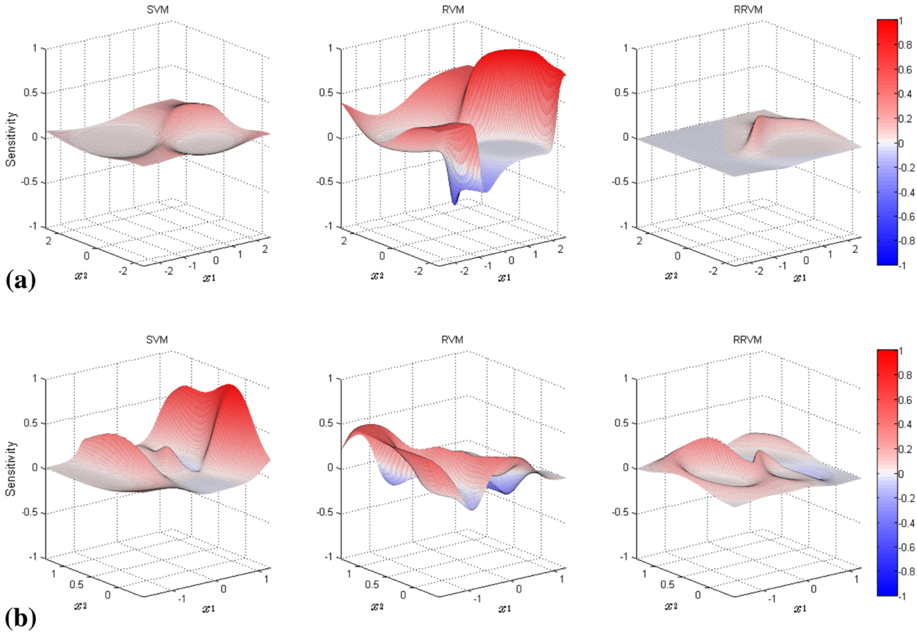


Fig. 7 The sensitivity curves of SVM, RVM, and RRVM obtained from **a** the simulated data in Fig. 1 and **b** the contaminated Ripley data in Fig. 4

5 Conclusion

A robust RVM which is insensitive to outliers far away from their own classes is developed in this paper. The proposed method utilizes a weighting strategy, assigning small weights to outliers. Given a prior distribution of weights, the weight values are determined in a probabilistic way and computed automatically during training. The variational inference method is employed to estimate the posterior distribution over unknown model coefficients including the weights. Consequently, no validation set is needed to optimize model parameters and all available data can be utilized for model training. Also, it is shown that such weighting approach makes a logistic loss function bounded. In other words, a loss value corresponding to an outlier is limited to an upper bound so that the outlier cannot further influence a decision boundary. In addition, a guideline for determining hyperparameter values is discussed. The experimental results show that the proposed method performs consistently better than other classification methods if a training data set is contaminated by outliers. In terms of optimizing model parameters, the proposed method is efficient for moderate-size data sets since it can avoid a grid search such as a cross validation method.

Since there is a large possibility that data sets obtained from real applications contain outliers, the proposed method in this study enable one to obtain robust models in practice. Valuable areas of future research may include the development of fast training algorithms for the proposed methods to deal with large data sets because the proposed iterative training algorithms contain a matrix inverse computation whose time complexity is cubic with respect to the number of observations.

Acknowledgements The authors thank the anonymous reviewers and editors for their helpful and constructive comments that greatly contributed to improving the paper.

Appendix 1: Proof of Proposition 1

The weight value $\mathbb{E}(w)$ is computed as the mean of $Gamma\left(w|\tilde{c},\tilde{d}\right)$, that is

$$\mathbb{E}(w)=\frac{\tilde{c}}{\tilde{d}}=\frac{c}{d-\ln\mathbb{E}\left(h\left(\boldsymbol{\beta},\xi\right)\right)}.$$

Since the logistic loss function is equivalent to a negative log likelihood function, the weighted logistic loss function can be written as

$$\begin{aligned}\mathbb{E}(w)l\left\{(2t-1)\,\boldsymbol{\beta}^T\boldsymbol{\phi}(\mathbf{x})\right\}&=-\mathbb{E}(w)\ln p\left(t|\boldsymbol{\beta}\right)\leq-\mathbb{E}(w)\ln\mathbb{E}\left(h\left(\boldsymbol{\beta},\xi\right)\right)\\&=-\frac{c\ln\mathbb{E}\left(h\left(\boldsymbol{\beta},\xi\right)\right)}{d-\ln\mathbb{E}\left(h\left(\boldsymbol{\beta},\xi\right)\right)}\leq c.\end{aligned}$$

Thus, the weighted logistic loss function is bounded by c.

Appendix 2: Proof of Proposition 2

Recall that $p\left(t|\boldsymbol{\beta}\right)\geq h\left(\boldsymbol{\beta},\xi\right)$. Taking the expectations on both sides with respect to $\boldsymbol{\beta}$ yields the following result:

$$\mathbb{E}\left(p\left(t|\boldsymbol{\beta}\right)\right)=p\left(t|\hat{\boldsymbol{\beta}}\right)\geq\mathbb{E}\left(h\left(\boldsymbol{\beta},\xi\right)\right)$$

where $\hat{\boldsymbol{\beta}}$ denotes the expectation of $\boldsymbol{\beta}$. Since $0\leq p\left(t|\hat{\boldsymbol{\beta}}\right)\leq 1$, it is always true that $\mathbb{E}\left(h\left(\boldsymbol{\beta},\xi\right)\right)\leq 1\Leftrightarrow\ln\mathbb{E}\left(h\left(\boldsymbol{\beta},\xi\right)\right)\leq 0$.

($\Rightarrow$) If $\ln\mathbb{E}\left(h\left(\boldsymbol{\beta},\xi_i\right)\right)$ is 0, then the weight $\mathbb{E}(w)$ should be 1 since $\ln\mathbb{E}\left(h\left(\boldsymbol{\beta},\xi_i\right)\right)$ approaches to 0 as $p\left(t|\hat{\boldsymbol{\beta}}\right)$ goes to 1. Therefore, if

$$0\leq\mathbb{E}(w)=\frac{c}{d-\ln\mathbb{E}\left(h\left(\boldsymbol{\beta},\xi\right)\right)}\leq 1$$

then c should be equal to d.

($\Leftarrow$) If $c=d\equiv r$, then

$$0\leq\mathbb{E}(w)=\frac{r}{r-\ln\mathbb{E}\left(h\left(\boldsymbol{\beta},\xi\right)\right)}\leq 1$$

since $\ln\mathbb{E}\left(h\left(\boldsymbol{\beta},\xi\right)\right)$ is always negative.

References

An, L. T. H., & Tao, P. D. (1997). Solving a class of linearly constrained indefinite quadratic problems by D.C. algorithms. *Journal of Global Optimization*, *11*(3), 253–285.

Bishop, C. M., & Tipping, M. E. (2000), Variational relevance vector machine. In *Proceedings of the 16th conference on uncertainty in artificial intelligence* (pp. 46–53).

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*(2), 121–167.

Caruana, R., & Niculescu-Mizil, A. (2004). Data mining in metric space: An empirical analysis of supervised learning performance criteria. In *Proceedings of the 10th international conference on knowledge discovery and data mining* (pp. 69–78).

Chang, C. C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 27:21–27:27.

Christmann, A., & Steinwart, I. (2004). On robustness properties of convex risk minimization methods for pattern recognition. *Journal of Machine Learning Research*, *5*, 1007–1034.

Debruyne, M., Serneels, S., & Verdonck, T. (2009). Robustified least squares support vector classification. *Journal of Chemometrics*, *23*(9), 479–486.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Fang, Y., & Jeong, M. K. (2008). Robust probabilistic multivariate calibration model. *Technometrics*, *50*, 305–316.

Frank, A., & Asuncion, A. (2010). *UCI machine learning repository*. Irvine, CA: University of California, School of Information and Computer Science.

Hwang, S., Yum, B., & Jeong, M. K. (2014). Robust relevance vector machine with variational inference for improving virtual metrology accuracy. *IEEE Transaction on Semiconductor Manufacturing*, *27*, 1–12.

Hwang, S., Kim, N., Jeong, M. K., & Yum, B. (2015). Robust kernel based regression with bounded influence for outliers. *Journal of Operations Research Society* (to appear).

Jaakkola, T. S. (2000). *Tutorial on variational approximation methods*. Technical Report, MIT Artificial Intelligence Lab.

Jaakkola, T. S., & Jordan, M. I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing*, *10*(1), 25–37.

Lee, K., Kim, N., & Jeong, M. K. (2014). A sparse signomial model for classification and regression. *Annals of Operations Research*, *216*, 257–286.

Lin, X. W., Wahba, G., Xiang, D., Gao, F. Y., Klein, R., & Klein, B. (2000). Smoothing spline ANOVA models for large data sets with Bernoulli observations and the randomized GACV. *Annals of Statistics*, *28*(6), 1570–1600.

Ling, C. X., Huang, J., & Zhang, H. (2003). AUC: A better measure than accuracy in comparing learning algorithms. In *Proceedings of the 2003 Canadian artificial intelligence conference* (pp. 329–341).

Ma, Z., & Leijon, A. (2011). Bayesian estimation of beta mixture models with variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(11), 2160–2173.

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

Neal, R. M. (1996). *Bayesian learning for neural networks*. New York: Springer.

Ormerod, J. T., & Wand, M. P. (2010). Explaining variational approximations. *The American Statistician*, *64*(2), 140–153.

Park, S. Y., & Liu, Y. (2011). Robust penalized logistic regression with truncated loss functions. *Canadian Journal of Statistics*, *39*(2), 300–323.

Ratsch, G., Onoda, T., & Muller, K. R. (2001). Soft margins for AdaBoost. *Machine Learning*, *42*(3), 287–320.

Song, Q., Hu, W., & Xie, W. (2002). Robust support vector machine with bullet hole image classification. *IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews*, *32*(4), 440–448.

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

Wu, Y., & Liu, Y. (2007). Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, *102*(479), 974–983.