Soft Comput (2006) 10: 219–223
DOI 10.1007/s00500-005-0474-z

ORIGINAL PAPER

S. Wang · J. Zhu · F. L. Chung · Hu Dewen

Experimental study on parameter choices in norm-r support vector regression machines with noisy input

Published online: 27 April 2005

Abstract In [1], with the evidence framework, the almost inversely linear dependency between the optimal parameter r in norm-r support vector regression machine r-SVR and the Gaussian input noise is theoretically derived. When r takes a non-integer value, r-SVR cannot be easily realized using the classical QP optimization method. This correspondence attempts to achieve two goals: (1) The Newton-decent-method based implementation procedure of r-SVR is presented here; (2) With this procedure, the experimental studies on the dependency between the optimal parameter r in r-SVR and the Gaussian noisy input are given. Our experimental results here confirm the theoretical claim in [1].

Keywords Support vector regression (SVR) · r-loss functions · Newton descent method

1 Introduction

Support vector regression machines [2, 3, 6–8] have earned their success in various applications, such as pattern recognition, modeling and data mining. Since noise often occurs in real input dataset, it is very important for us to derive the dependency between the optimal parameters in support vector regression machines and the noisy input. Kwok and Tsang [4], Smola et al. [5] and Smola and Schölkopf [6] established the linear dependency between ε in ε-SVR with the ε-insensitive loss function and the noisy input. In [1], the authors investigated theoretically optimal parameter choices for Huber-SVR with the Huber loss functions and r-SVR with the norm-r loss functions. Our theoretical results claim that the parameter μ in Huber-SVR has the linear dependency with the noisy input, and the parameter r in r-SVR has the inversely linear dependency with the input noise.

However, when r takes a non-integer value, r-SVR cannot be easily realized using the classical QP optimization method. Due to this fact, unlike ε-SVR and Huber-SVR, to date, the experimental study on the optimal parameter choice for r-SVR with the noisy input has not yet been reported. In this correspondence, the Newton-descent-method based implementation approach on r-SVR is derived. With this approach, the experimental study on the inversely linear dependency between the optimal parameter r in r-SVR and the Gaussian noisy input is organized in the paper. Our experimental results here support the theoretical claim in [1].

S. Wang (✉)
School of Information, Southern Yangtz University, Wuxi, China
E-mail: wxwangst@yahoo.com.cn

J. Zhu · S. Wang
Department of Computer, Nanjing University of Science and Technology, Nanjing, China

S. Wang · F. L. Chung
Department of Computing, HongKong Polytechnic University, HongKong, China

H. Dewen
School of Automation, National Defense University of Science and Technology, ChangSha, China

S. Wang
Labarotory of Computer Science, Institute of Software, Chinese Academy of Science, China

2 Norm-r support vector regression machine r-SVR

Assume there is a dataset D with n-dimensional input vector $\boldsymbol{x}$

$$D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}, \quad \mathbf{x} \in R^n, \quad y \in R,$$

where l denotes the number of samples in D, we are interested in obtaining a weight vector $\mathbf{w}$ and an parameter b such that

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

can be used to approximate the dataset well. In [1], we define the so-called norm-r loss function as

$$\mathbf{L}_r (f(\mathbf{x}) - y) = |f(\mathbf{x}) - y|^r,$$

according to the definition in [1], r-SVR is equivalent to the following minimization problem:

$$\min \Phi(\mathbf{w}, \xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i,$$

s.t.

$$\begin{cases} |f(\mathbf{x}_i) - y_i|^r \leqslant \xi_i, \quad i = 1, \ldots, l \\ \xi_i \geqslant 0, \end{cases} \tag{1}$$

where C is a predefined constant, and $\xi_i (i = 1, 2, \ldots, l)$ measures the upper amount that $|f(\mathbf{x}_i) - y_i|^r$ must satisfy.

By forming the Lagrangian, (1) can be transformed into the corresponding dual quarantic programming problem. Its derivation is derived as follows. With $\alpha_i, \alpha_i^*, \gamma_i$ being the Lagrangian multipliers, we introduce the Lagirangian $L(\mathbf{w}, b, \xi, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \gamma)$ below:

$$\begin{aligned} \min \mathbf{L}\left(\mathbf{w}, b, \xi, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \gamma\right) &= \frac{1}{2}\|\mathbf{w}\|^2 \\ &+ C\sum_i \xi_i + \sum_i \alpha_i \left[y_i - f(\mathbf{x}_i) - \xi_i^{\frac{1}{r}}\right] \\ &+ \sum_i \alpha_i^* \left[f(\mathbf{x}_i) - y_i - \xi_i^{\frac{1}{r}}\right] - \sum_i \gamma_i \xi_i \end{aligned}$$

s.t.

$$\begin{cases} \alpha_i \geqslant 0, \\ \alpha_i^* \geqslant 0, \\ \gamma_i \geqslant 0, \ i = 1, \ldots, l, \\ \xi_i \geqslant 0 \end{cases} \tag{2}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots\ldots, \alpha_l)$, $\boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*, \ldots\ldots, \alpha_l^*)$.

The KKT optimality conditions require that at optimality the gradients of the Lagrangian vanish, that is to say,

$$\begin{aligned} &\frac{\partial L}{\partial \mathbf{w}} = 0, \\ &\frac{\partial L}{\partial b} = 0, \\ &\frac{\partial L}{\partial \xi_i} = 0, \\ &\frac{\partial L}{\partial \gamma_i} = 0, \end{aligned} \tag{3}$$

where $i = 1, \ldots, l$. After substituting (3) into (2) and further simplification, we can obtain the dual optimization problem of r-SVR:

$$\begin{aligned} \min L(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) &= \frac{1}{2}\sum_{i,j=1}^{l} \left(\alpha_i - \alpha_i^*\right)\left(\alpha_j - \alpha_j^*\right)\left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle \\ &- \sum_{i=1}^{l} \left(\alpha_i - \alpha_i^*\right) y_i + \frac{r-1}{r^{\frac{r}{r-1}} C^{\frac{1}{r-1}}} \sum_{i=1}^{l} \left(\alpha_i + \alpha_i^*\right)^{\frac{r}{r-1}} \end{aligned}$$

s.t.

$$\begin{cases} \alpha_i \geqslant 0, \\ \alpha_i^* \geqslant 0, \quad i = 1, \ldots, l \\ \\ \sum_{i=1}^{l} \left(\alpha_i - \alpha_i^*\right) = 0 \end{cases} \tag{4}$$

Remark 1 *In (4), we may replace $\left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle$ by some kernel function $K\left(\mathbf{x}_i, \mathbf{x}_j\right)$. In terms of the theoretical derivations in [1], kernel functions do not give a big impact on the optimal parameter choice for r-SVR, so, here we do not intend to extend our discussion into the kernel cases.*

Remark 2 *When no noise or very small noise exist in input datasets or the influence of noise may be ignored, we generally take $r = 2$, due to its simplicity ((4) may be easily solved by the classical QP method) and human being's habit. In fact, r-SVR is a generalized version of the often-used 2-SVR. However, it is easily seen from (4) that when r is a non-integer value, r-SVR cannot be solved using the classical QP method. Therefore, we must attempt to find out the alternative method. Fortunately, the following Newton-descent-method can help us to achieve this goal.*

Remark 3 *When an input dataset contains noise, we should choose an optimal parameter r in r-SVR to reduce the influence of noise well. Theoretical analysis in [1] shows that parameter r in r-SVR has the inversely linear dependency with noisy input. That is to say, with the increase of noise, parameter r in r-SVR should be linearly reduced to suppress the influence of noise. However, experimental studies were not reported in [1].*

3 Newton descent method for r-SVR

In the previous section, our discussion resulted in a constrained optimisation problem, which cannot be solved using the classical QP method. In the following, we address this optimisation problem (4) using the Newton descent method.

Define the following penalty term:

$$\begin{aligned} \mathbf{P}\left(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*\right) = -N &\left(\sum_{i=1}^{l} \log \alpha_i + \sum_{i=1}^{l} \log \alpha_i^*\right) \\ &+ M\left[\sum_{i=1}^{l} \left(\alpha_i - \alpha_i^*\right)\right]^2, \end{aligned}$$

where M, N denote the penalty factors in the above penalty term. Furthermore, let

$$\varphi\left(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*\right) = \mathbf{L}\left(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*\right) + \mathbf{P}\left(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*\right)$$

thus, we can transform the constrained optimization problem (4) into the following unconstrainted optimization one:

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} \varphi\left(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*\right) = \mathbf{L}\left(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*\right) + \mathbf{P}\left(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*\right). \tag{5}$$

According to the Newton descent method, we can use the following [6] to calculate $\boldsymbol{\alpha}, \boldsymbol{\alpha}^*$:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{\alpha}_k \\ \boldsymbol{\alpha}_k^* \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\alpha}_{k-1} \\ \boldsymbol{\alpha}_{k-1}^* \end{bmatrix} \\ &- \left[\nabla^2 \varphi\left(\boldsymbol{\alpha}_{k-1}, \boldsymbol{\alpha}_{k-1}^*\right)\right]^{-1} \nabla \varphi\left(\boldsymbol{\alpha}_{k-1}, \boldsymbol{\alpha}_{k-1}^*\right), \end{aligned} \tag{6}$$

where k denotes the iterative number, and the matrix $\nabla\varphi(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*)$ is defined as

$$\nabla\varphi\left(\alpha,\alpha^*\right)=\begin{bmatrix}\frac{\partial\varphi}{\partial\alpha_1}\\ \vdots\\ \frac{\partial\varphi}{\partial\alpha_l}\\ \frac{\partial\varphi}{\partial\alpha_1^*}\\ \vdots\\ \frac{\partial\varphi}{\partial\alpha_l^*}\end{bmatrix},$$

where

$$\frac{\partial\varphi}{\partial\alpha_i}=\sum_{j=1}^{l}\left(\alpha_j-\alpha_j^*\right)\left\langle\mathbf{x}_i,\mathbf{x}_j\right\rangle\ -y_i+\left(\frac{\alpha_i+\alpha_i^*}{rC}\right)^{\frac{1}{r-1}}$$
$$-\frac{N}{\alpha_i}+2M\sum_{j=1}^{l}\left(\alpha_j-\alpha_j^*\right),\quad i=1,\ldots,l,$$

$$\frac{\partial\varphi}{\partial\alpha_i^*}=-\sum_{j=1}^{l}\left(\alpha_j-\alpha_j^*\right)\left\langle\mathbf{x}_i,\mathbf{x}_j\right\rangle+y_i+\left(\frac{\alpha_i+\alpha_i^*}{rC}\right)^{\frac{1}{r-1}}$$
$$-\frac{N}{\alpha_i^*}-2M\sum_{j=1}^{l}\left(\alpha_j-\alpha_j^*\right),\quad i=1,\ldots,l$$

and the $2l\times 2l$ matrix $\nabla^2\varphi\left(\boldsymbol{\alpha},\boldsymbol{\alpha}^*\right)$ is defined as

$$\nabla^2\varphi\left(\boldsymbol{\alpha},\boldsymbol{\alpha}^*\right)=\begin{bmatrix}\frac{\partial^2\varphi}{\partial\alpha_1^2}&\cdots&\frac{\partial^2\varphi}{\partial\alpha_1\partial\alpha_l}&\frac{\partial^2\varphi}{\partial\alpha_1\partial\alpha_1^*}&\cdots&\frac{\partial^2\varphi}{\partial\alpha_1\partial\alpha_l^*}\\ \vdots&\ddots&\vdots&\vdots&\ddots&\vdots\\ \frac{\partial^2\varphi}{\partial\alpha_l\partial\alpha_1}&\cdots&\frac{\partial^2\varphi}{\partial\alpha_l^2}&\frac{\partial^2\varphi}{\partial\alpha_l\partial\alpha_1^*}&\cdots&\frac{\partial^2\varphi}{\partial\alpha_l\partial\alpha_l^*}\\ \frac{\partial^2\varphi}{\partial\alpha_1^*\partial\alpha_1}&\cdots&\frac{\partial^2\varphi}{\partial\alpha_1^*\partial\alpha_l}&\frac{\partial^2\varphi}{\partial\left(\alpha_1^*\right)^2}&\cdots&\frac{\partial^2\varphi}{\partial\alpha_1^*\partial\alpha_l^*}\\ \vdots&\ddots&\vdots&\vdots&\ddots&\vdots\\ \frac{\partial^2\varphi}{\partial\alpha_l^*\partial\alpha_1}&\cdots&\frac{\partial^2\varphi}{\partial\alpha_l^*\partial\alpha_l}&\frac{\partial^2\varphi}{\partial\alpha_l^*\partial\alpha_1^*}&\cdots&\frac{\partial^2\varphi}{\partial\left(\alpha_l^*\right)^2}\end{bmatrix},$$

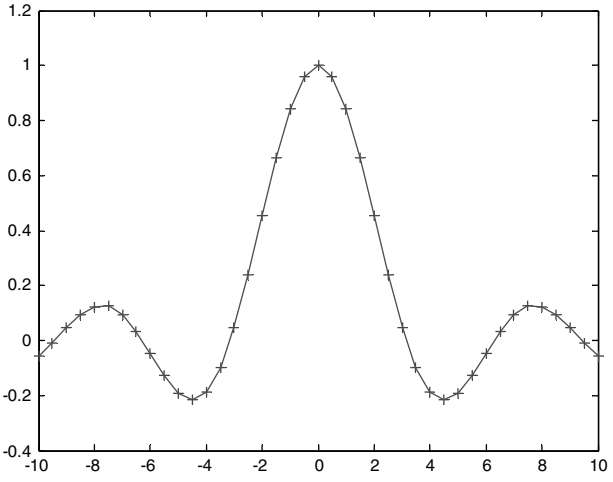

Fig. 1 Regression curve of r-SVR with $r = 2$

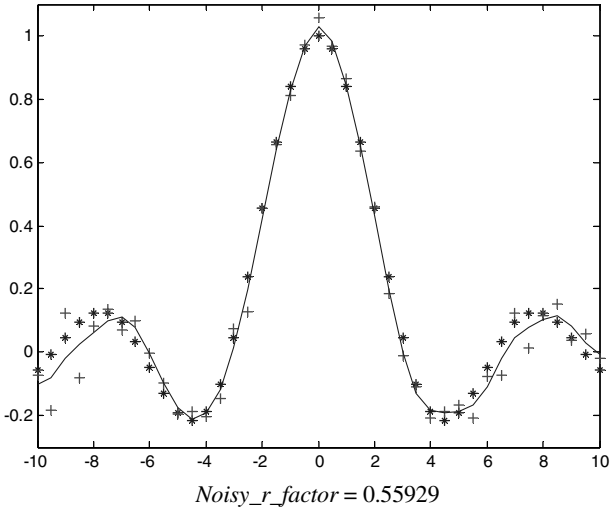

Fig. 2 Regression curve of r-SVR with $r = 2$ for noisy input with $k = 0.05$

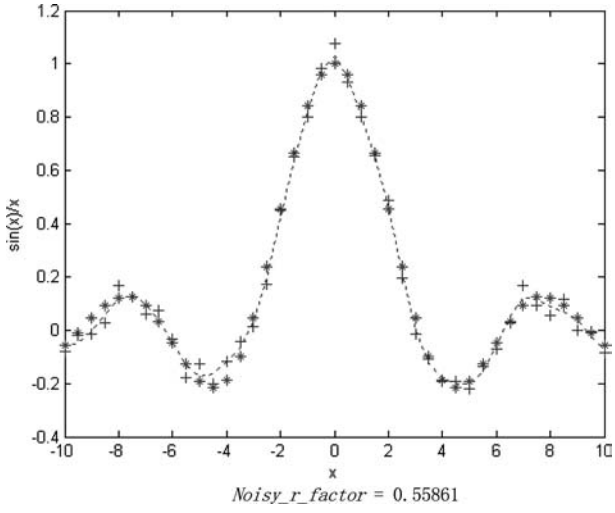

Fig. 3 Regression curve of r-SVR with $r = 1.8$ for noisy input with $k = 0.05$

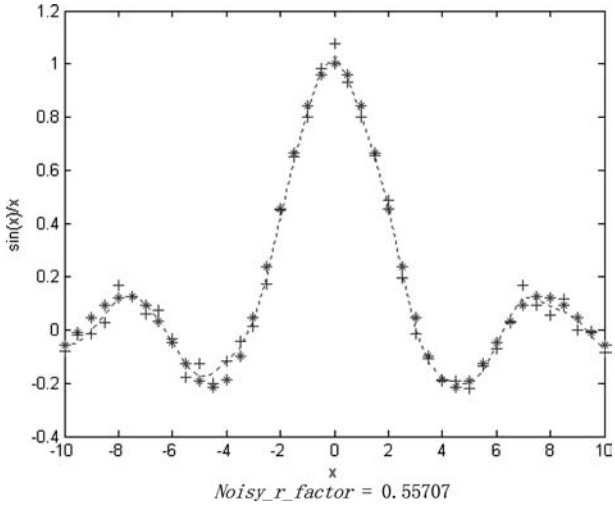

Fig. 4 Regression curve of r-SVR with$r = 1.5$ for noisy input with $k = 0.05$

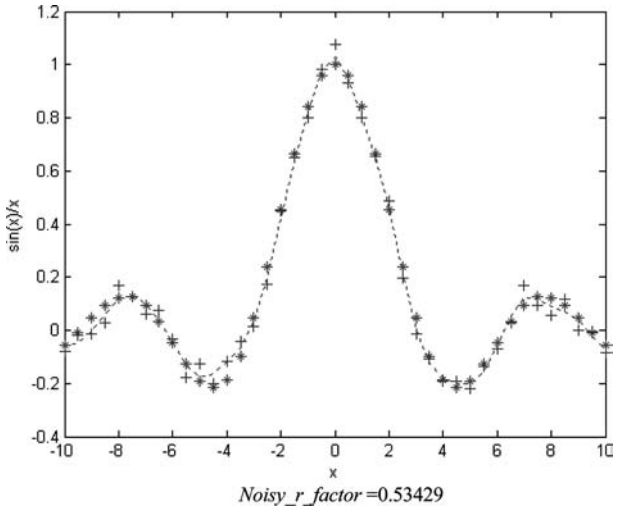

Fig. 5 Regression curve of r-SVR with $r = 1.3$ for noisy input with $k = 0.05$

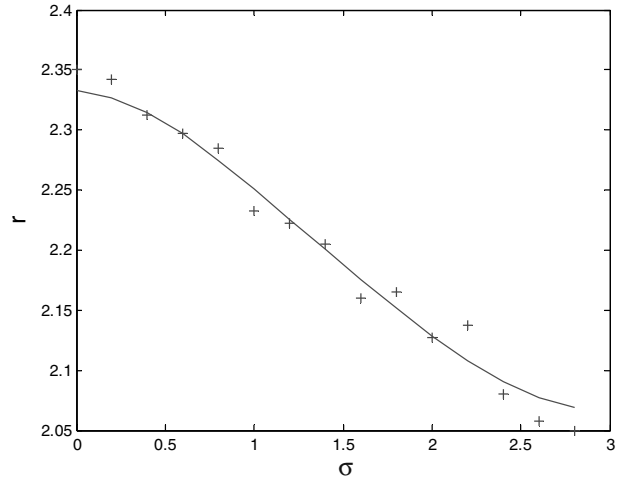

Fig. 6 The relationship between r and σ when $k = 0.05$

where

$$\frac{\partial^2\varphi}{\partial\alpha_i\partial\alpha_j} = \begin{cases} \langle \mathbf{x}_i, \mathbf{x}_i\rangle + \frac{1}{r-1}\left(\frac{\alpha_i+\alpha_i^*}{C}\right)^{\frac{2-r}{r-1}} + \frac{N}{\alpha_i^2} + 2M, & i = j \\ \langle \mathbf{x}_i, \mathbf{x}_j\rangle + 2M, & i \neq j \end{cases}, i, j = 1, \ldots, l$$

$$\frac{\partial^2\varphi}{\partial\alpha_i^*\partial\alpha_j^*} = \begin{cases} \langle \mathbf{x}_i, \mathbf{x}_i\rangle + \frac{1}{r-1}\left(\frac{\alpha_i+\alpha_i^*}{C}\right)^{\frac{2-r}{r-1}} + \frac{N}{\alpha_i^2} + 2M, & i = j \\ \langle \mathbf{x}_i, \mathbf{x}_j\rangle + 2M, & i \neq j \end{cases}, i, j = 1, \ldots, l$$

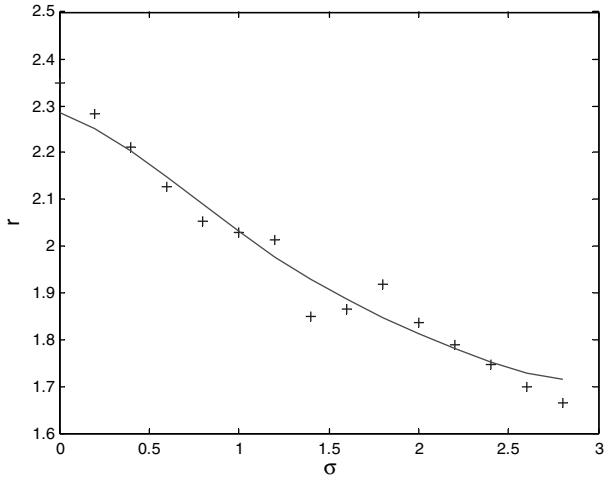

Fig. 7 The relationship between r and σ when $k = 0.15$

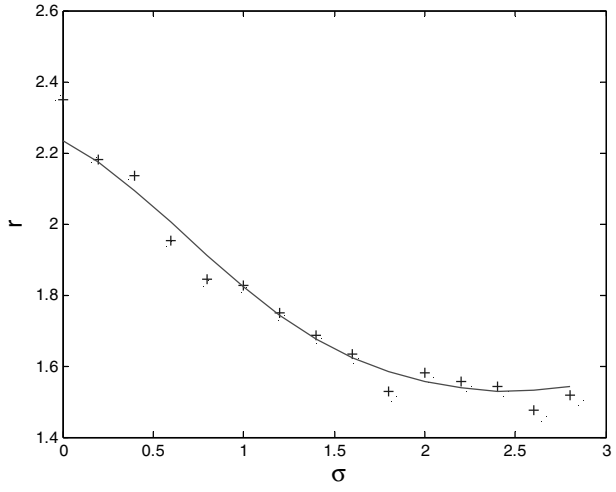

Fig. 8 The relationship between r and σ when $k = 0.30$

$$\frac{\partial^2\varphi}{\partial\alpha_i\partial\alpha_j^*} = \frac{\partial^2\varphi}{\partial\alpha_i^*\partial\alpha_j}$$

$$= \begin{cases} \langle \mathbf{x}_i, \mathbf{x}_i\rangle + \frac{1}{r-1}\left(\frac{\alpha_i+\alpha_i^*}{C}\right)^{\frac{2-r}{r-1}} + \frac{N}{\alpha_i^2} + 2M, & i = j \\ \langle \mathbf{x}_i, \mathbf{x}_j\rangle + 2M, & i \neq j \end{cases}, i, j = 1, \ldots, l$$

Thus, given appropriate M and N, we can obtain the approximate solution of (4) by running [6] repeatedly.

4 Experimental studies

In [1], the authors theoretically derived the almost inversely linear dependency between r and σ. Now, in order to further confirm this theoretical result, let us do experimental studies by applying the above algorithm to a benchmarking function Sin $C(x)$.

Given a Sin C function $y = \sin(x)/x, x \in [-10, 10]$, let us generate its uniformly distributed dataset $(x_i, y_i), i =$

$1, \ldots, 41$, with x varying from -10 to 10 using the step length 0.5. First, for the dataset, we can easily construct its regression curve $f = f(x)$ using r-SVR. Next, in order to investigate the dependency relationship between r and the noisy input, let $y' = \sin(x)/x + k \cdot \eta, x \in [-10, 10]$, where k is a noise-signal ratio and $\eta \sim N(0, \sigma)$ represents the Gaussian noise. Similarly, we can generate its corresponding sampling dataset $(x_i, y_i'), i = 1, \ldots, 41$, and obtain its r-SVR regression curve $f' = f(x)$. In order to make the experimental results fair, we take σ from $[0, 2.8]$, and use the Gaussian noise distribution to generate 15 groups of the corresponding sampling datasets for each given σ. For each given σ, we take r as the average result of all 20 r values which can minimize $\sum_{i=1}^{41} | f_i' - f_i |$ respectively for each group of the sampling datasets.

Our experimental results can be seen from Figs. 1–8. Figure 1 demonstrates the real output of $y = \sin(x)/x$ and its regression curve using r-SVR, where "+" denotes a sample $(x_i, y_i), i = 1, \ldots, 41$, and the solid line corresponds to its r-SVR regression curve. Figures 2–5 show the r-SVR regression curves for the noisy input (k =0.05) with r=2, 1.8, 1.5, 1.1, where "*" corresponds to a noisy version of sample "+". In order to measure the role of the parameter r, we define

$$Noisy\_r\_factor \approx 1 - \frac{\sum_i (f(x_i) - y_i)^2}{\sum_i \left(f'(x_i) - y_i'\right)^2} \tag{7}$$

where $f(x)$denotes the regression curve for the dataset, and $f'(x)$denotes the regression curve for the noisy dataset. Obviously, we should take such an r that *Noisy_r_factor* reaches its possible minimum. In other words, the smaller *Noisy_r_factor* is, the better the corresponding regression curve $f'(x)$is. From Figs. 2–5, we easily see that r =1.3 is a good choice (the corresponding *Noisy_r_factor* =0.53429). Figures 6, 7 and 8 depict the dependency relationships between $r$and $\sigma$for all 15 σvalues with different k(see "+" in the figures), where we use the curves to roughly indicate the change tendencies between $r$and σ, respectively. We can easily see from these figures that when noise is very small, i.e, k and σ is small, r may be roughly taken as 2, which is very good in line with the fact that in most cases, we use 2-SVR for realistic datasets without noise With the increase of k and/or σ, there obviously exist the almost inversely linear dependency between $r$and σ. However, when k and/or σ become comparatively larger, i.e, the datasets are seriously distorted, the inversely linear relationship between $r$and σdoes not exist anymore (see Fig. 8). In other words, r-SVR may become ineffective for seriously distorted datasets. In summary, the above experimental results validate the theoretical claim on optimal parameter choice of r-SVR with noisy input.

5 Conclusion

When r takes a non-integer value, realizing r-SVR is not a trivial task. This correspondence presents the Newton-decent-method based implementation procedure of r-SVR. With this procedure, the experimental studies on the dependency relationship between the optimal parameter r in r-SVR and the Gaussian noisy input are given. Our experimental results here confirm the theoretical analysis in [1]. That is to say, there exists the inversely linear dependency between the optimal parameter r in r-SVR and the noisy input.

Acknowledgements This work is supported by the RGC Competitive Earmarked Research Grant (grant No. PolyU 5065/98E), Natural Science Foundation of China (grant No. 60225015), Natural Science Foundation of JiangSu Province (grant No. BK2003017), Excellent Young Teacher Fund of Education Ministry of China, National Key Lab. of Novel Software Technologies at NanJing University and Lab of Computer Science, Institute of Software, Chinese Academy of Science, China.

References

1. Wang S, Zhu J et al Theoretically optimal parameter choices for support vector regression machines with noisy input. J Soft Computing (accepted)
2. Cristianini N, Shawe-Taylor J (2000) An Introduction to support vector machines. Cambridge University Press, Cambridge
3. Vapnik V (1998) Statistical learning theory. Wiley, New York
4. Kwok JT, Tsang IW (2003) Linear dependency between ε and the input noise in ε-support vector regression. IEEE Trans Neural Networks, **5**:544–553
5. Smola AJ, Murata N, Schölkopf B, Müller K-R (1998) Asymptotically optimal choice of ε-loss for support vector machines. In: Proceedings of the international conference on artificial neural networks[C]
6. Smola AJ, Schölkopf, B (1998) A tutotial on support vector regression. Royal Holloway College, NeuroCOLT2 Technical Report NC2-TR-1998-030
7. Law MH, Kwok JT (2001) Bayesian support vector regression. In: Proceedings of the English international workshop on artificial intelligence and statistics. Key West, Florida: 239–244
8. Gao JB, Gunn SR, Ham's CJ (2002) A probabilistic framework for SVM regression and error bar estimation. Machine Learning, **46**:71–89
9. Vladimir Cherkassky, Yunqian Ma (2004) Practical selection of SVM parameters and noise estimation for SVM regression. Neural Networks. **17** (1):113–126