Journal of the Korean Data & Information Science Society
2015, **26**(6), 1537–1545
http://dx.doi.org/10.7465/jkdi.2015.26.6.1537
한국데이터정보과학회지

SVQR with asymmetric quadratic loss function†

Jooyong Shim[1] · Malsuk Kim[2] · Kyungha Seok[3]

[13]Department of Statistics, Inje University
[2]Department of Social Welfare & Childcare, Yeungnam University College
Received 18 August 2015, revised 18 November 2015, accepted 23 November 2015

Abstract

Support vector quantile regression (SVQR) can be obtained by applying support vector machine with a check function instead of an e-insensitive loss function into the quantile regression, which still requires to solve a quadratic program (QP) problem which is time and memory expensive. In this paper we propose an SVQR whose objective function is composed of an asymmetric quadratic loss function. The proposed method overcomes the weak point of the SVQR with the check function. We use the iterative procedure to solve the objective problem. Furthermore, we introduce the generalized cross validation function to select the hyper-parameters which affect the performance of SVQR. Experimental results are then presented, which illustrate the performance of proposed SVQR.

Keywords: Asymmetric quadratic loss function, generalized cross validation.

1. Introduction

Quantile regression has been a popular method for estimating the quantiles of a conditional distribution on the values of input variables since Koenker and Basset (1978) introduced linear quantile regression. Just as classical linear regression methods based on minimizing sum of squared residuals enable us to estimate a wide variety of models for conditional mean functions, quantile regression methods offer a mechanism for estimating models for the full range of conditional quantile functions, including the conditional median function. By supplementing the estimation of conditional mean functions with techniques for estimating an entire family of conditional quantile functions, quantile regression is capable of providing a better statistical analysis of the stochastic relationships among random variables. An introduction to, and look at current research areas of quantile regression can be found in Koenker and Hallock (2001) and Yu *et al.* (2003).

† This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012000646) and (2011-0009705).
[1] Adjunct professor, Institute of statistical Information, Department of Statistics, Inje University, Kimhae 621-749, Korea.
[2] Assistant professor, Department of Social Welfare & Childcare, Yeungnam University College, Daegu 705-703, Korea.
[3] Corresponding author: Professor, Institute of statistical Information, Department of Statistics, Inje University, Kimhae 621-749, Korea. E-mail:statskh@inje.ac.kr

Support vector machine (SVM) is used as a new technique for regression and classification problems. The SVM is based on the structural risk minimization (SRM) principle, which has been shown to be superior to the traditional empirical risk minimization (ERM) principle. SRM minimizes an upper bound on the expected risk, unlike ERM, which minimizes the error on the training data. By minimizing this bound, high generalization performance can be achieved. In particular, for the SVM regression case, SRM results in regularized ERM with e-insensitive loss function. Introductions to and overviews of recent developments of SVM and kernel machines can be found in Vapnik (1995, 1998), Smola and Scholkopf (1998), Shim and Hwang (2010) and Wang (2005).

The minimization problem associated with linear quantile regression is in essence the linear programming (LP) optimization problem, which is based on simplex algorithm or interior point algorithm. The current state of algorithms for nonlinear quantile regression is far less satisfactory. The widely used algorithm is interior point algorithm. Nonlinear quantile regression poses new algorithmic challenge. Refer to Koenker and Park (1996) and Koenker and Hallock (2001) for the algorithms. Training an SVM requires the solution to a quadratic programming (QP) optimization problem. But QP problem presents some inherent limitations which results in computational difficulty especially for the large data sets. Platt (1998), Flake and Lawrence (2002) developed the sequential minimal optimization (SMO) algorithm which divides the QP problem into a series of small QP problems to avoid such computational difficulty. Perez-Cruz *et al.* (2000) proposed IRWLS algorithm for SVR by transforming the Lagrangian function into sum of quadratic terms by defining associated weights of predicted errors.

Among kernel machines, least squares support vector machine (LS-SVM, Suykens and Vandewalle, 1999) has been proved to be a very appealing and promising method. Solving nonlinear modeling by convex optimization without suffering from many local minima like SVM is one of its strong points. In addition, LS-SVM uses the liner equation which is simple to solve and good for computational time saving. Many tests and comparisons showed great performance of LS-SVM on several benchmark data set problems and are applicable to various types of data. Introductions to and overviews of recent developments of LS-SVM can be found in Suykens and Vandewalle (1999) and Shim and Seok (2014).

SVQR can be obtained by applying SVM with a check function instead of an e-insensitive loss function into the quantile regression, which still requires to solve a quadratic program (QP) problem which is time and memory expensive.

In this paper we use the asymmetric quadratic loss function instead of the check function used in SVQR, which leads the fast computation. In Section 2 we briefly review SVQR with quadratic programming. In Section 3 we propose the SVQR with asymmetric quadratic loss function and introduce the generalized cross validation function (GCV) to select the hyper-parameters. In Section 4 and 5 we perform numerical studies through artificial examples and give the conclusions, respectively.

2. Support vector quantile regression

Let the training data set denoted by $(\boldsymbol{x}_i, y_i)_{i=1}^n$, with each input $\boldsymbol{x}_i \in R^d$ and the response $y_i \in R$, where the output variable y_i is related to the input vector $\boldsymbol{x}_i$. Here the feature mapping function $\phi(\cdot) : R^d \to R^{d_f}$ maps the input space to the higher dimensional feature space where the dimension d_f is defined in an implicit way. An inner product in feature space has an equivalent kernel in input space, $\phi(\boldsymbol{x}_i)'\phi(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ (Mercer, 1909). Several choices of the kernel $K(\cdot,\cdot)$ are possible. We consider the nonlinear regression case, in which the quantile regression function $q(\boldsymbol{x}_i)$ of the response given $\boldsymbol{x}_i$ can be regarded as a nonlinear function of input vector $\boldsymbol{x}_i$ such as $q_\theta(\boldsymbol{x}_i) = \boldsymbol{\omega}'\phi(\boldsymbol{x}_i) + b$.

With a check function $\rho_\theta(\cdot)$, the estimator of the θth quantile regression function can be defined as any solution to the optimization problem,

$$\min\ \ell(q_\theta|\boldsymbol{x}) = \sum_{i=1}^{n} \rho_\theta(y_i - q(\boldsymbol{x}_i))$$

where the check function $\rho_\theta(r) = \theta r I_{(r>=0)} + (1-\theta) r I_{(r<0)}$.

We can express the regression problem by formulation for SVM as follows.

$$\min\ L = \frac{1}{2}\boldsymbol{w}'\boldsymbol{w} + C\sum_{i=1}^{n}(\theta\xi_i + (1-\theta)\xi_i^*)$$

subject to

$$y_i - \boldsymbol{w}'\phi(\boldsymbol{x}_i) - b \le \xi_i,\ \boldsymbol{w}'\phi(\boldsymbol{x}_i) + b - y_i \le \xi_i^*,\ \xi_i, \xi_i^* \ge 0,$$

where C is a positive regularization parameter penalizing the training errors.

We construct a Lagrange function as follows:

$$\begin{aligned} L =& \frac{1}{2}\boldsymbol{w}'\boldsymbol{w} + C\sum_{i=1}^{n}(\theta\xi_i + (1-\theta)\xi_i^*) - \sum_{i=1}^{n}\alpha_i(\xi_i - y_i + \boldsymbol{w}'\phi(\boldsymbol{x}_i) + b) \\ & - \sum_{i=1}^{n}\alpha_i^*(\xi_i^* + y_i - \boldsymbol{w}'\phi(\boldsymbol{x}_i) - b) - \sum_{i=1}^{n}(\eta_i\xi_i + \eta_i^*\xi_i^*). \end{aligned} \tag{2.1}$$

We notice that the positivity constraints $\alpha_i, \alpha_i^*, \eta_i, \eta_i^* \ge 0$ should be satisfied. After taking partial derivatives of equation (2.1) with regard to the primal variables $(\boldsymbol{w}, b, \xi_i, \xi_i^*)$ and plugging them into equation (2.1), we have the optimization problem with $\phi(\boldsymbol{x}_i)'\phi(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ (Mercer, 1909) below.

$$\max\ -\frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)y_i \tag{2.2}$$

with constraints

$$0 \le \alpha_i \le \theta C,\ 0 \le \alpha_i^*(1-\theta)C \text{ and } \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0.$$

Solving the above equation (2.2) with the constraints determines the optimal Lagrange multipliers, α_i, α_i^*, the estimator of the θth quantile regression function given the input vector $\boldsymbol{x}_t$ is obtained as follows:

$$\widehat{q}_\theta(\boldsymbol{x}_t) = \sum_{i=1}^{n} K(\boldsymbol{x}_t, \boldsymbol{x}_i)(\widehat{\alpha}_i - \widehat{\alpha}_i^*) + \widehat{b}. \tag{2.3}$$

Here $\widehat{b}$ is obtained via Kuhn-Tucker conditions (Kuhn and Tucker, 1951) such as,

$$\widehat{b} = \frac{1}{n_s}\sum_{i\in I_s}(y_i - K(\boldsymbol{x}_i, \boldsymbol{x})(\widehat{\boldsymbol{\alpha}} - \widehat{\boldsymbol{\alpha}}^*)), \tag{2.4}$$

where $\widehat{\boldsymbol{\alpha}} = (\widehat{\alpha}_1, \cdots, \widehat{\alpha}_n)'$, $\widehat{\boldsymbol{\alpha}}^* = (\widehat{\alpha}_1^*, \cdots, \widehat{\alpha}_n^*)'$ and n_s is the size of the set $I_s = \{i = 1, \cdots, n| 0 < \widehat{\alpha}_i < C\theta, 0 < \widehat{\alpha}_i^* < C(1-\theta)\}$.

The functional structures of SVQR in (2.3) is characterized by the hyper-parameters, C and the kernel parameters. To select the hyper-parameters of SVQR we consider the cross validation (CV) function as follows:

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\rho_\theta(y_i - \widehat{q}_\theta(\boldsymbol{x}_i)^{(-i)}), \tag{2.5}$$

where λ is the set of hyper-parameters and $\widehat{q}_\theta(\boldsymbol{x}_i)^{(-i)}$ is the quantile regression function estimated without ith observation. Since for each candidates of parameters, $\widehat{q}_\theta(\boldsymbol{x}_i)^{(-i)}$ for $i = 1, \cdots, n$, should be evaluated, selecting parameters using CV function is computationally formidable. Yuan (2006) proposed the generalized approximate cross validation (GACV) function to select the set of hyper-parameters λ for SVQR as follows:

$$GACV(\lambda) = \frac{\sum_{i=1}^{n}\rho_\theta(y_i - \widehat{q}_\theta(\boldsymbol{x}_i))}{n - trace(H)},$$

where H is the hat matrix such that $\widehat{q}(\theta|\boldsymbol{x}) = H\boldsymbol{y}$ with the (i, j)th element $h_{ij} = \frac{\partial \widehat{q}_\theta(\boldsymbol{x}_i)}{\partial y_j}$. From Li *et al.* (2007) we have that the trace of the hat matrix H equals to the size of set I_s used in (2.4).

3. SVQR with asymmetric quadratic loss function

With each input vector $\boldsymbol{x}_i \in R^d$ and the response $y_i \in R$, the θth quantile regression function can be defined as any solution to the optimization problem,

$$\min\ \ell(q_\theta|\boldsymbol{x}) = \sum_{i=1}^{n}\rho_\theta(y_i - q(\boldsymbol{x}_i)).$$

Here the check function in $\rho_\theta(\cdot)$ is defined as $\rho_\theta(r) = \theta r I_{(r>0)} + (1-\theta) r I_{(r<0)}$, which can be written as asymmetric quadratic loss function as follows:

$$\rho_\theta(r_i) = \frac{\theta}{|r_i|}I(r_i > 0)r_i^2 + \frac{(1-\theta)}{|r_i|}I(r_i < 0)r_i^2. \tag{3.1}$$

Using (3.1) we can express the regression problem by formulation for weighted LS-SVM as follows:

$$\min\ L = \frac{1}{2}\boldsymbol{w}'\boldsymbol{w} + C\sum_{i=1}^{n}u_i(\theta)e_i^2$$

over $\{\boldsymbol{w}, \boldsymbol{e}\}$ subject to equality constraints,

$$e_i = y_i - \boldsymbol{\omega}'\phi(\boldsymbol{x}_i) - b,\ i = 1, \cdots, n.,$$

where $u_i(\theta) = \frac{\theta}{|e_i|} I(e_i > 0) + \frac{(1-\theta)}{|e_i|} I(e_i < 0)$ and C is a positive regularization parameter penalizing the training errors.

The Lagrangian function can be constructed as

$$L = \frac{1}{2}\boldsymbol{w}'\boldsymbol{w} + \frac{C}{2}\sum_{i=1}^{n} u_i(\theta)e_i^2 + \sum_{i=1}^{n} \alpha_i(y_i - \boldsymbol{w}'\phi(x_i) - b - e_i), \tag{3.2}$$

where α_i's are Lagrange multipliers. The Karush-Kuhn-Tucker (Smola and Scholkopf, 1998) conditions for optimality are given by

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{w}} = 0 \quad &\rightarrow \quad \boldsymbol{w} - \sum_{i=1}^{n} \alpha_i \phi(x_i) = 0 \\
\frac{\partial L}{\partial b} = 0 \quad &\rightarrow \quad \sum_{i=1}^{n} \alpha_i = 0 \\
\frac{\partial L}{\partial e_i} = 0 \quad &\rightarrow \quad C u_i(\theta) e_i - \alpha_i = 0,\ i = 1, \cdots, n, \\
\frac{\partial L}{\partial \alpha_i} = 0 \quad &\rightarrow \quad y_i - \boldsymbol{w}'\phi(\boldsymbol{x}_i) - e_i = 0,\ i = 1, \cdots, n.
\end{aligned} \tag{3.3}$$

From the equations (3.2), (3.3) and the application of Mercer's conditions (Mercer, 1909) the optimal values of α_i 's and $\widehat{b}$ are obtained from the linear equations:

$$\begin{pmatrix} K + U^{-1}/C & \mathbf{1} \\ \mathbf{1}' & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ b \end{pmatrix} = \begin{pmatrix} \boldsymbol{y} \\ 0 \end{pmatrix}, \tag{3.4}$$

where $K = (K(\boldsymbol{x}_i, \boldsymbol{x}_j))_{n \times n}$, U is the diagonal matrix of $u_i(\theta)$'s and $\boldsymbol{y} = (y_1, \cdots, y_n)'$.

Since $u_i(\theta)$ in (3.4) contains (α_i, b), we need to apply iterative procedure which starts with initialized value of $U = I$. In each iteration step, ($\widehat{\alpha}_i, \widehat{b}$) is obtained from the linear equations as follows:

$$\begin{pmatrix} \widehat{\boldsymbol{\alpha}} \\ \widehat{b} \end{pmatrix} = \begin{pmatrix} (K+U^{-1}/C)^{-1} - (K+U^{-1}/C)^{-1}\mathbf{1}(\mathbf{1}'(K+U^{-1}/C)^{-1}\mathbf{1})^{-1}\mathbf{1}'(K+U^{-1}/C)^{-1} \\ (\mathbf{1}'(K+U^{-1}/C)^{-1}\mathbf{1})^{-1}\mathbf{1}'(K+U^{-1}/C)^{-1} \end{pmatrix} \boldsymbol{y}$$

where U is the diagonal matrix of $u_i(\theta)$'s computed by the optimal values of α_i's and $\widehat{b}$ obtained in the previous iteration step.

The estimator of the θth quantile regression function given the input vector $\boldsymbol{x}_t$ is obtained as follows:

$$\widehat{q}_\theta(\boldsymbol{x}_t) = \sum_{i=1}^{n} K(\boldsymbol{x}_t, \boldsymbol{x}_i)\widehat{\alpha}_i + \widehat{b}.$$

To select the hyper-parameters of sparse SVQR with asymmetric quadratic loss function we consider the cross validation (CV) function as follows:

$$CV(\boldsymbol{\lambda}) = \frac{1}{n}\sum_{i=1}^{n} u_i(\theta)(y_i - \widehat{q}_\theta^{(-i)}(\boldsymbol{x}_i|\boldsymbol{\lambda}))^2,$$

instead of CV function used in SVQR with check function (2.5), where $\boldsymbol{\lambda}$ is a set of hyper-parameters and $\widehat{q}_\theta^{(-i)}(\boldsymbol{x}_i|\boldsymbol{\lambda})$ is the θth quantile regression function estimated without ith observation. Since for each candidates of hyper- parameters, $\widehat{q}_\theta^{(-i)}(\boldsymbol{x}_i|\boldsymbol{\lambda})$ for $i = 1, \cdots, n$, should be evaluated, selecting hyper- parameters using CV function is computationally formidable. By using leaving- out-one lemma (Craven and Wahba, 1979) the ordinary cross validation (OCV) function can be obtained as

$$OCV(\boldsymbol{\lambda}) = \frac{1}{n}\sum_{i=1}^{n} u_i(\theta)\left(\frac{y_i - \widehat{q}_\theta(\boldsymbol{x}_i|\boldsymbol{\lambda})}{1 - \dfrac{\partial \widehat{q}_\theta(\boldsymbol{x}_i|\boldsymbol{\lambda})}{\partial y_i}}\right)^2 = \frac{1}{n}\sum_{i=1}^{n} u_i(\theta)\left(\frac{y_i - \widehat{q}_\theta(\boldsymbol{x}_i|\boldsymbol{\lambda})}{1 - h_{ii}}\right)^2$$

where H is the hat matrix such that $\widehat{q}_\theta(\boldsymbol{x}|\boldsymbol{\lambda}) = H\boldsymbol{y}$ with the (i,j)th element $h_{ij} = \partial \widehat{q}_\theta(\boldsymbol{x}_i)/\partial y_j$. Here the hat matrix can be expressed as follows:

$$H = (K, \mathbf{1})\begin{pmatrix} (K+U^{-1}/C)^{-1} - (K+U^{-1}/C)^{-1}\mathbf{1}(\mathbf{1}'(K+U^{-1}/C)^{-1}\mathbf{1})^{-1}\mathbf{1}'(K+U^{-1}/C)^{-1} \\ (\mathbf{1}'(K+U^{-1}/C)^{-1}\mathbf{1})^{-1}\mathbf{1}'(K+U^{-1}/C)^{-1} \end{pmatrix}$$

where U is the final estimate. Replacing h_{ii} by their average $tr(H)/n$, the generalized cross validation (GCV) function can be obtained as

$$GCV(\boldsymbol{\lambda}) = \frac{n\sum_{i=1}^{n} u_i(\theta)(y_i - \widehat{q}_\theta(\boldsymbol{x}_i|\boldsymbol{\lambda}))^2}{(n - tr(H))^2}.$$

4. Numerical studies

In this section, we illustrate the performance of the proposed quantile regression estimation with asymmetric quadratic loss function through the simulated examples on the nonlinear quantile regression case. We generate 100 data sets of size 200 in a similar manner to Cawley *et al.* (2004). The univariate input observations x follows a uniform distribution(0,2), the corresponding responses y are generated as follows:

$$y = \sin(2\pi x) + \sigma(x)\epsilon,$$

where $\epsilon \sim \chi_{(2)} - 2$ and $\sigma(x) = 0.5\sqrt{2.1 - x}$. The Gaussian kernel function is utilized in this example, which is

$$K(x_1, x_2) = \exp\left(-\frac{1}{\sigma^2}||x_1 - x_2||^2\right).$$

Figure 4.1 shows a family of quantile functions estimated by with asymmetric quadratic loss function (SVQR_AQ) and support vector quantile regression with check function (SVQR_C) for the training data set. The estimated quantile regression functions for $\theta = 0.1, 0.5, 0.9$ are superimposed on the scatter plots. In SVQR_AQ with the values of C and σ^2 are chosen by GACV function such as (300, 1.5) for $\theta = 0.1$, (400, 2) for $\theta = 0.5$, and (400, 2) for $\theta = 0.9$. In SVQR_C the values of C and σ^2 are chosen by GCV function such as (400, 1.5) for $\theta = 0.1$, (300, 2) for $\theta = 0.5$, and (300, 1.5) for $\theta = 0.9$. As seen from Figure 4.1, in both procedures the three estimated quantile regression functions reflect well the heteroscedastic structure of the error term. They have their (local) minima and (local) maxima at different x values. For example, the 0.1th, 0.5th and 0.9th quantile regression functions have maxima at x =0.25 and 1.25, respectively, and minima at 0.75 and 1.75, respectively. To illustrate the estimation performance of SVQR_AQ, we compare it with SVQR_QP via 100 data sets, where the mean squared error (PMSE) is used as the performance measure defined by

$$MSE = \frac{1}{200}\sum_{i=1}^{200}(\widehat{q}_\theta(x_i) - q_\theta(x_i))^2 \text{ for } \theta = 0.1, 0.5, 0.9.$$

The averages of 100 MSEs from SVQR_AQ and SVQR_C are obtained in Table 4.1. We can see that both procedures have almost same for the estimation performance.

Table 4.1 Comparison of MSEs for 100 simulated data sets (standard error in parenthesis)

θ	SVQR_AQ			SVQR_C		
	0.1	0.5	0.9	0.1	0.5	0.9
	0.0154 (0.0009)	0.0555 (0.0034)	0.3428 (0.0251)	0.0153 (0.0021)	0.0549 (0.0033)	0.3436 (0.0253)

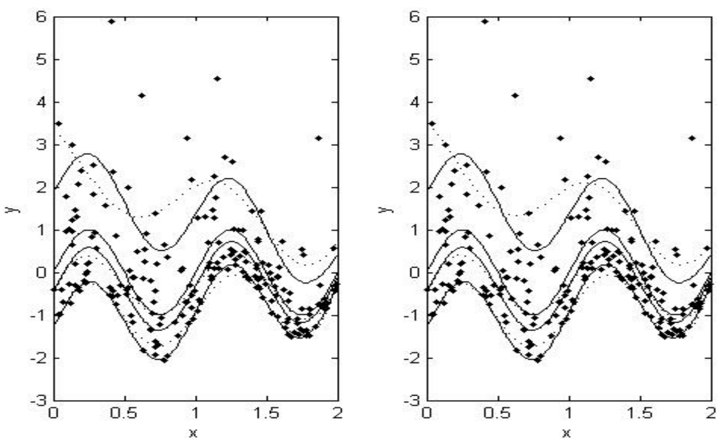

Figure 4.1 An illustration of SVQR_AQ (Left) and SVQR_C (Right) for a data set of size 200 generated from the process. True quantile regression function (solid line) and the estimated quantile regression function (dotted line) for $\theta = 0.1, 0.5, 0.9$ are superimposed on the scatter plots.

With 100 simulated data sets, CPU-times of SVQR_AQ are compared with that of SVQR_C computed by the built-in function of MATLAB. Here (C, σ^2) are fixed as (100, 1). Table 4.2 shows CPU-times in seconds of both procedures (run MATLAB R2006b over Core(TM) at

3.6GHz) on 100 data sets with different sample sizes (n=300, 500, 700, 1000). From table 4.2 we can see that SVQR_AQ is much faster than SVQR_C, which implies that the proposed procedure is appropriate procedure for the large training data sets.

Table 4.2 Average CPU times of training SVQR_AQ and SVQR_C (standard error in parenthesis)

θ	n	SVQR_AQ	SVQR_C	θ	n	SVQR_AQ	SVQR_C
0.1	300	0.4373 (0.0038)	0.7780 (0.0062)	0.1	700	3.5217 (0.0228)	9.5533 (0.0214)
	500	1.5762 (0.0133)	2.6008 (0.0102)		1000	8.9920 (0.0486)	25.5814 (0.0334)
0.5	300	0.4345 (0.0038)	0.8184 (0.0055)	0.5	700	3.5456 (0.237)	10.0324 (0.0235)
	500	1.6165 (0.0120)	2.2732 (0.0114)		1000	9.0062 (0.0459)	26.7567 (0.0463)
0.9	300	0.4463 (0.0031)	0.7563 (0.0053)	0.9	700	3.5353 (0.0198)	9.3658 (0.0194)
	500	1.5960 (0.0114)	2.5405 (0.0104)		1000	8.9443 (0.0472)	25.2274 (0.0314)

5. Conclusions

In this paper, we dealt with estimating the nonlinear quantile regression function by SVQR with asymmetric quadratic loss function and obtained GCV function for the proposed procedure. Through the examples we showed that the proposed procedure derives the satisfying solutions. We also found that SVQR with asymmetric quadratic loss function is much faster than SVQR with check function, which implies that the proposed procedure is appropriate for the large training data sets.

References

Cawley, G. C., Talbot, N. L. C., Foxall, R. J., Dorling, S. R. and Mandic, D. P. (2004). Heteroscedastic kernel ridge regression. *Neurocomputing*, **57**, 105-124.

Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerical Mathematics*, **31**, 377-403.

Flake, G. W. and Lawrence, S. (2002). Efficient SVM regression training with SMO. *Machine Learning*, **46**, 271-290.

Koenker, R. and Bassett, G. (1978). Regression quantile. *Econometrica*, **46**, 33-50.

Koenker, R. and Hallock, K. F. (2001). Quantile regression. *Journal of Economic Perspectives*, **40**, 122-142.

Koenker, R. and Park, B. J. (1996). An interior point algorithm for nonlinear quantile regression. *Journal of Econometrics*, **71**, 265-283.

Kuhn, H. W. and Tucker, A. W. (1951). Nonlinear programming. In *proceedings of 2nd Berkeley symposium* . Berkeley: University of California Press, 481-492.

Li, Y., Liu, Y. and Zhu, J. (2007). Quantile regression in reproducing kernel Hilbert spaces. *Journal of the American Statistical Association*, **102**, 255-268.

Mercer, J. (1909). Functions of positive and negative type and their Connection with theory of integral equations. *Philosophical Transactions of Royal Society A*, 415-446.

Perez-Cruz, F., Navia-Vazquez, A., Alarcon-Diana, P. L. and Artes-Rodriguez, A. (2000). An IRWLS procedure for SVR. *In proceedings of European association for signal processing*, EUSIPO 2000, Tampere, Finland.

Platt, J. (1998). *Sequential minimal optimization: A fast algorithm for training support vector machines*, Microsoft Research Technical Report, CA. MSR-TR-98-14.

Shim, J. and Hwang, C. (2010). Semiparametric support vector machine for accelerated failure time model. *Journal of the Korean Data & Information Science Society*, **21**, 467-477.

Shim, J. and Seok, K. (2014). A transductive least squares support vector machine with the difference convex algorithm. *Journal of the Korean Data & Information Science Society*, **25**, 455-464.

Smola, A. and Scholkopf, B. (1998). On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica*, **22**, 211-231.

Suykens, J. A. K. and Vandewalle, J. (1999). Least square support vector machine classifier. *Neural Processing Letters*, **9**, 293-300.

Vapnik, V. N. (1995). *The nature of statistical learning theory*, Springer, New York.

Vapnik, V. N. (1998). *Statistical Learning Theory* . John Wiley, New York.

Wang, L. (Ed.) (2005). *Support vector machines : Theory and application*, Springer, Berlin Heidelberg New York.

Yu, K., Lu, Z. and Stander, J. (2003). Quantile regression: Applications and current research area. *The Statistician*, **52**, 331-350.

Yuan, M. (2006). GACV for quantile smoothing splines. *Computational Statistics and Data Analysis*, **50**, 813-829.