# Sparse SVQR for Detecting Differently Expressed Genes 

Jooyong Shim*<br>Department of Data Science, Inje University, Gimhae 621-749, Korea<br>(Received April 10, 2014; Revised May 15, 2014; Accepted May 20, 2014)


#### Abstract

Support vector quantile regression (SVQR) is capable of providing more complete description of the linear and nonlinear relationships among random variables. In this paper we propose the sparse SVQR whose objective function is composed of a weighted quadratic loss function and $l_{1}$ norm penalty term. We use the iterative reweighted least squares (IRWLS) procedure to solve the objective problem of the proposed SVQR. Furthermore, we introduce the generalized approximate cross validation function to select the hyper-parameters which affect the performance of SVQR. Experimental results are then presented, which illustrate the performance of the sparse SVQR using IRWLS procedure.


Key words: Generalized cross validation function, Iterative reweighted least squares procedure, Quadratic loss function, Sparsity, Support vector quantile regression

## 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 Bassett [1] 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 [2-5].

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 [6-8].

Sparsity is known as an important feature of kernel regression models. It provides efficiency in predicting the regression function, which implies that the predicted regression function of the test data can be obtained with the small number of data in the training data set. SVM provides sparsity in which the number of support vectors depends on the number of training data and the size of insensitivity. A small number of support vectors implies sparsity of the model. Tipping [9] proposed a Bayesian approach referred to as the relevance vector machine, providing more sparsity. However the relevance vector machine has computational problems since there are no closed-form solutions for maximizing the marginal likelihood.

SVQR can be obtained by applying SVM with a check func-

[^0]tion instead of an e-insensitive loss function into the quantile regression $[10,11]$. But SVQR does not provide sparsity due to zero insensitiveness of check function. Here we define support vectors as the index numbers corresponding to nonzero Lagrange multiplier differences.

In this paper we use the weighted quadratic loss function and $l_{1}$ norm penalty term instead of the check function and $l_{2}$ norm penalty term used in SVQR, which leads the fast computation and the sparsity of the model. In Section 2 we briefly review SVQR using quadratic programming. In Section 3 we propose the sparse SVQR which uses iterative reweighted least squares procedure. In Section 4 and 5 we perform numerical studies through artificial examples and give the conclusions, respectively.

## Support Vector Quantile Regression

Let the training data set denoted by $\left(x_{i}, y_{i}\right)_{i=1}^{n}$, 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 $\phi(\cdot): R^{d} \rightarrow R^{d_{f}}$ 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 [12], $\phi\left(\boldsymbol{x}_{i}\right)^{\prime} \phi\left(\boldsymbol{x}_{j}\right)=K\left(\boldsymbol{x}_{i}, \boldsymbol{x}_{j}\right)$. Several choices of the kernel $K(\cdot, \cdot)$ are possible. We consider the nonlinear regression case, in which the quantile regression function $q\left(\boldsymbol{x}_{\boldsymbol{i}}\right)$ of the response given $\boldsymbol{x}_{i}$ can be regarded as a nonlinear function of input vector $\boldsymbol{x}_{i}$ such as $q_{\theta}\left(\boldsymbol{x}_{i}\right)=\omega^{\prime} \phi\left(\boldsymbol{x}_{i}\right)+b$.
With a check function $\rho_{\theta}(\cdot)$, the estimator of the $\theta$ th quantile regression function can be defined as any solution to the optimization problem,

$$
\begin{equation*}
\min l\left(q_{\theta} \mid \boldsymbol{x}\right)=\sum_{i=1}^{n} \rho_{\theta}\left(y_{i}-q\left(\boldsymbol{x}_{i}\right)\right) \tag{1}
\end{equation*}
$$

where $\rho_{\theta}(r)=\theta r I_{(r \geq 0)}+(1-\theta) r I_{(r<0)}$.
We can express the regression problem by formulation for SVM as follows.

$$
\begin{equation*}
\min L=\frac{1}{2} \boldsymbol{w}^{\prime} \boldsymbol{w}+C \sum_{i=1}^{n}\left(\theta \xi_{i}+(1-\theta) \xi_{i}^{*}\right) \tag{2}
\end{equation*}
$$

subject to

$$
y_{i}-\boldsymbol{w}^{\prime} \phi\left(\boldsymbol{x}_{i}\right)-b \leq \xi_{i}, \boldsymbol{w}^{\prime} \phi\left(\boldsymbol{x}_{i}\right)+b-y_{i} \leq \xi_{i}^{*}, \xi_{i}, \xi_{i}^{*} \geq 0
$$

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

We construct a Lagrange function as follows:

$$
\begin{align*}
L= & \frac{1}{2} \boldsymbol{w}^{\prime} \boldsymbol{w}+C \sum_{i=1}^{n}\left(\theta \xi_{i}+(1-\theta) \xi_{i}^{*}\right)-\sum_{i=1}^{n} \alpha_{i}\left(\xi_{i}-y_{i}+\boldsymbol{w}^{\prime} \phi\left(\boldsymbol{x}_{i}\right)+b\right) \\
& -\sum_{i=1}^{n} \alpha_{i}^{*}\left(\xi_{i}^{*}+y_{i}-\boldsymbol{w}^{\prime} \phi\left(\boldsymbol{x}_{i}\right)-b\right)-\sum_{i=1}^{n}\left(\eta_{i} \xi_{i}+\eta_{i}^{*} \xi_{i}^{*}\right) . \tag{3}
\end{align*}
$$

We notice that the positivity constraints $\alpha_{i}, \alpha_{i}^{*}, \eta_{i}, \eta_{i}^{*} \geq 0$ should be satisfied. After taking partial derivatives of (3) with regard to the primal variables $\left(\boldsymbol{w}, b, \xi_{i}, \xi_{i}^{*}\right)$ and plugging them into (3), we have the optimization problem with $\phi\left(\boldsymbol{x}_{i}\right)^{\prime} \phi\left(\boldsymbol{x}_{j}\right)=$ $K\left(x_{i}, x_{j}\right)$ below.

$$
\begin{equation*}
\max -\frac{1}{2} \sum_{i, j=1}^{n}\left(\alpha_{i}-\alpha_{i}^{*}\right)\left(\alpha_{j}-\alpha_{j}^{*}\right) K\left(\boldsymbol{x}_{i}, \boldsymbol{x}_{j}\right)+\sum_{i=1}^{n}\left(\alpha_{i}-\alpha_{i}^{*}\right) y_{i} \tag{4}
\end{equation*}
$$

with constraints

$$
0 \leq \alpha_{i} \leq \theta C, 0 \leq \alpha_{i}^{*}(1-\theta) C \text { and } \sum_{i=1}^{n}\left(\alpha_{i}-\alpha_{i}^{*}\right)=0
$$

Solving the above equation with the constraints determines the optimal Lagrange multipliers, $\alpha_{i}, \alpha_{i}^{*}$, the estimator of the $\theta$ th quantile regression function given the input vector $\boldsymbol{x}_{t}$ is obtained as follows:

$$
\begin{equation*}
\hat{q}_{\theta}\left(\boldsymbol{x}_{t}\right)=\sum_{i=1}^{n} K\left(\boldsymbol{x}_{t}, \boldsymbol{x}_{i}\right)\left(\hat{\alpha}_{i}-\hat{\alpha}_{i}^{*}\right)+\hat{b} . \tag{5}
\end{equation*}
$$

Here $\hat{b}$ is obtained via Kuhn-Tucker conditions [13] such as,

$$
\begin{equation*}
\hat{b}=\frac{1}{n_{s}} \sum_{i \in I_{s}}\left(y_{i}-K\left(\boldsymbol{x}_{i}, \boldsymbol{x}\right)\left(\hat{\boldsymbol{\alpha}}-\hat{\boldsymbol{\alpha}}^{*}\right)\right), \tag{6}
\end{equation*}
$$

where $\hat{\boldsymbol{\alpha}}=\left(\hat{\alpha}_{1}, \cdots, \hat{\alpha}_{n}\right)^{\prime}, \hat{\boldsymbol{\alpha}}^{*}=\left(\hat{\alpha}_{1}^{*}, \cdots, \hat{\alpha}_{n}^{*}\right)^{\prime}$ and $n_{s}$ is the size of the set $I_{s}=\left\{i=1, \cdots, n \mid 0<\hat{\alpha}_{i}<C \theta, 0<\hat{\alpha}_{1}^{*}<C(1-\theta)\right\}$.

The functional structures of SVQR 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:

$$
\begin{equation*}
C V(\lambda)=\frac{1}{n} \sum_{i=1}^{n} \rho_{\theta}\left(y_{i}-\hat{q}_{\theta}\left(x_{i}\right)^{(-i)}\right), \tag{7}
\end{equation*}
$$

where $\lambda$ is the set of hyper-parameters and $\hat{q}_{\theta}\left(x_{i}\right)^{(-i)}$ is the quantile regression function estimated without $i$ th observation. Since for each candidates of parameters, $\hat{q}_{\theta}\left(\boldsymbol{x}_{i}\right)^{(-i)}$ for $i=1, \cdots, n$, should be evaluated, selecting parameters using CV function is computationally formidable. Yuan [14] proposed the generalized approximate cross validation (GACV) function to select the set of hyper-parameters $\lambda$ for SVQR as follows:

$$
\begin{equation*}
\operatorname{GACV}(\lambda)=\frac{\sum_{i=1}^{n} \rho_{\theta}\left(y_{i}-\hat{q}_{\theta}\left(\boldsymbol{x}_{i}\right)\right)}{n-\operatorname{trace}(H)} \tag{8}
\end{equation*}
$$

where $H$ is the hat matrix such that $\hat{q}(\theta \mid \boldsymbol{x})=H \boldsymbol{y}$ with the $(i, j)$ th element $h_{i j}=\frac{\partial \hat{q}_{\theta}\left(\boldsymbol{x}_{i}\right)}{\partial y_{j}}$. From Li, Liu and Zhu [15] we have that the trace of the hat matrix $H$ equals to the size of set $I_{s}$ used in (6).

## Sparse SVQR Using IRWLS Procedure

With each input vector $x_{i} \in R^{d}$ and the response $y_{i} \in R$, we consider the regression model to estimate the quantile regression function,

$$
\begin{equation*}
y_{i}=q_{\theta}\left(\boldsymbol{x}_{i}\right)+\varepsilon_{i}, i=1,2, \cdots, n \tag{9}
\end{equation*}
$$

where $\varepsilon_{i}$ is assumed to follow independently asymmetric Laplacian distribution $(\theta, 0, \sigma)$ whose probability density function is given as

$$
f(y)=\frac{\theta(1-\theta)}{\sigma} \exp \left(-\frac{1}{\sigma} \rho_{\theta}(y)\right)
$$

The negative log likelihood of the given data set can be expressed as (constant terms are omitted)

$$
\begin{equation*}
l\left(q_{\theta} \mid \boldsymbol{x}\right)=\frac{1}{\sigma} \sum_{i=1}^{n} \rho_{\theta}\left(y_{i}-q_{\theta}\left(\boldsymbol{x}_{i}\right)\right) \tag{10}
\end{equation*}
$$

We consider the nonlinear regression case, in which the quantile regression function $q\left(\boldsymbol{x}_{i}\right)$ of the response given $\boldsymbol{x}_{i}$ can be regarded as a nonlinear function of input vector $\boldsymbol{x}_{i}$ such as $q_{\theta}\left(\boldsymbol{x}_{i}\right)=K_{i} \boldsymbol{\alpha}+b$, where $K_{i}$ is the $i$ th row of $K$ and $(\boldsymbol{\alpha}, b)$ are the vector of $n$ weights and a bias. Then the maximum likelihood estimates of $(\boldsymbol{\alpha}, b)$ are obtained by minimizing the negative loglikelihood function,

$$
\begin{equation*}
l(\boldsymbol{\alpha}, b)=\sum_{i=1}^{n} \rho_{\theta}\left(y_{i}-K_{i} \boldsymbol{\alpha}-b\right) . \tag{11}
\end{equation*}
$$

The maximum likelihood estimates of ( $\boldsymbol{\alpha}, b$ ) generally lead severe overfitting, we are encouraged to use a prior over $\boldsymbol{\alpha}$. Then the penalized maximum likelihood estimates (the maximum a posteriori estimates) of ( $\boldsymbol{\alpha}, b$ ) are obtained by minimizing the objective function,

$$
\begin{equation*}
L(\boldsymbol{\alpha}, b)=l(\boldsymbol{\alpha}, b)+\log (p(\boldsymbol{\alpha})), \tag{12}
\end{equation*}
$$

where $p(\boldsymbol{\alpha})$ is some prior over $\boldsymbol{\alpha}$.
To have the sparsity on estimation of $\boldsymbol{\alpha}$, we use a Laplacian prior [16] such that

$$
p(\boldsymbol{\alpha}) \propto \exp \left(-\gamma\|\boldsymbol{\alpha}\|_{1}\right),
$$

where $\|\boldsymbol{\alpha}\|_{1}=\sum_{i=1}^{n}\left|\alpha_{i}\right|$ denotes $l_{1}$ norm and $\gamma$ is a positive constant.
The objective function can be rewritten as

$$
\begin{equation*}
L(\boldsymbol{\alpha}, b)=l(\boldsymbol{\alpha}, b)+\gamma\|\boldsymbol{\alpha}\|_{1} . \tag{13}
\end{equation*}
$$

Here $\gamma$ controls the tradeoff between the goodness-of-fit on the data and $\|\boldsymbol{\alpha}\|_{1}$.
The penalty term in objective function in (13) is not differentiable with respect to $\boldsymbol{\alpha}$ and $b$, we need a modification of $\|\boldsymbol{\alpha}\|_{1}$ for IRWLS procedure.
We define the objective function given $\boldsymbol{\alpha}^{*}$ as

$$
\begin{equation*}
L\left(\boldsymbol{\alpha}, b \mid \boldsymbol{\alpha}^{*}\right)=l(\boldsymbol{\alpha}, b)+\frac{\gamma}{2} \sum_{i=1}^{n}\left(\frac{\alpha_{i}^{2}}{\left|\alpha_{i}^{*}\right|}+\left|\alpha_{i}^{*}\right|\right), \tag{14}
\end{equation*}
$$

then $L\left(\boldsymbol{\alpha}, b \mid \boldsymbol{\alpha}^{*}\right) \geq L(\boldsymbol{\alpha}, b)$ with equality if and only if $\boldsymbol{\alpha}=\boldsymbol{\alpha}^{*}$ [17] but $l(\boldsymbol{\alpha}, b)$ is not yet differentiable with respect to $\boldsymbol{\alpha}$ and $b$.

We modify $l(\boldsymbol{\alpha}, b)$ with the weighted quadratic loss function such as

$$
l(\boldsymbol{\alpha}, b)=\sum_{i=1}^{n} w t_{i}(\theta)\left(y_{i}-K_{i} \boldsymbol{\alpha}-b\right)^{2}
$$

where $w t_{i}(\theta)=\frac{\theta}{\left|r_{i}\right|} I\left(r_{i} \geq 0\right)+\frac{(1-\theta)}{\left|r_{i}\right|} I\left(r_{i}<0\right)$ and $r_{i}=y_{i}-K_{i} \boldsymbol{\alpha}-b$.
Then the objective function (14) can be rewritten as follows:

$$
\begin{equation*}
L\left(\boldsymbol{\alpha}, b \mid \boldsymbol{\alpha}^{*}\right)=\sum_{i=1}^{n} w t(\theta)\left(y_{i}-K_{i} \boldsymbol{\alpha}-b\right)^{2}+\frac{\gamma}{2} \sum_{i=1}^{n}\left(\frac{\alpha_{i}^{2}}{\left|\alpha_{i}^{*}\right|}+\left|\alpha_{i}^{*}\right|\right) . \tag{15}
\end{equation*}
$$

Now the objective function $L\left(\boldsymbol{\alpha}, b \mid \boldsymbol{\alpha}^{*}\right)$ is differentiable with respect to $\boldsymbol{\alpha}$ and $b$.

At $t$ th iteration of IRWLS procedure, we have

$$
\begin{align*}
L\left(\boldsymbol{\alpha}, b \mid \hat{\boldsymbol{\alpha}}^{(t)}, \hat{b}^{(t)}\right)= & \sum_{i=1}^{n} w t_{i}(\theta)\left(y_{i}-K_{i} \boldsymbol{\alpha}-b\right)^{2} \\
& +\frac{\gamma}{2} \sum_{i=1}^{n}\left(\frac{\alpha_{i}^{2}}{\left|\hat{\alpha}_{i}^{(t)}\right|}+\left|\hat{\alpha}_{i}^{(t)}\right|\right) \tag{16}
\end{align*}
$$

Then the estimates of ( $\boldsymbol{\alpha}, b$ ) are obtained by differentiating $L\left(\boldsymbol{\alpha}, b \mid \hat{\boldsymbol{\alpha}}^{(t)}, \hat{b}^{(t)}\right)$ with respect to $(\boldsymbol{\alpha}, b)$ as

$$
K W^{(t)}(\boldsymbol{y}-K \boldsymbol{\alpha}-b)+\gamma V^{(t)}=\mathbf{0} \text { and } \mathbf{1}^{\prime} W^{(t)}(\boldsymbol{y}-K \boldsymbol{\alpha}-b)=0
$$

where $W^{(t)}$ and $V^{(t)}$ are the diagonal matrices consisted of
$1 /\left|\hat{\alpha}_{i}^{(t)}\right|$ and $w t_{i}(\theta)$, respectively. Here $w t_{i}(\theta)=\frac{\theta}{\left|r_{i}\right|} I\left(r_{i} \geq 0\right)+$ $\frac{(1-\theta)}{\left|r_{i}\right|} I\left(r_{i}<0\right), r_{i}=y_{i}-K_{i} \hat{\boldsymbol{\alpha}}^{(t)}-\hat{b}^{(t)}$. Thus the estimates of $(\boldsymbol{\alpha}$, $b)$ are given as the solution of the linear equations:

$$
\binom{\hat{\boldsymbol{\alpha}}}{\hat{b}}=\left(\begin{array}{cc}
K W^{(t)} K+\gamma V^{(t)} & K W^{(t)} \mathbf{1}  \tag{17}\\
\mathbf{1} W^{(t)} K & \mathbf{1}^{\prime} W^{(t)} \mathbf{1}
\end{array}\right)^{-1}\binom{K W^{(t)}}{\mathbf{1}^{\prime} W^{(t)}} \boldsymbol{y}
$$

During iteration, we find that some $\alpha_{i}$ 's tend to zero keeping the value of objective function decreasing. This motivates that we can find sparse estimates of $\boldsymbol{\alpha}$ which provide decreasing value of the objective function at the same time.
We apply IRWLS procedure which starts with initialized values of $v=(1: n)^{\prime}$ and $\left(\hat{\boldsymbol{\alpha}}^{(0)}, \hat{b}^{(0)}\right)$ as follows:

1. Find $W^{(t)}$ and $V^{(t)}$ from $\left(\hat{\boldsymbol{\alpha}}(v)^{(t)}, \hat{b}^{(t)}\right)$.
2. Find $\left(\hat{\boldsymbol{\alpha}}(v)^{(t+1)}, \hat{b}^{(t+1)}\right)$ which minimizes $L\left(\boldsymbol{\alpha}(v), b \mid \hat{\boldsymbol{\alpha}}(v)^{(t)}\right.$, $\hat{b}^{(t)}$ ).
3. Set $\hat{\alpha}_{i}^{(t+1)}=0$ which is very close to zero.

Find $v=\left\{i \mid \hat{\alpha}_{i}^{(t+1)} \neq 0\right\}$.
4. Iterate 1-3 until $\left|L\left(\hat{\boldsymbol{\alpha}}(v)^{(t+1)}, \hat{b}^{(t+1)}\right)-L\left(\hat{\boldsymbol{\alpha}}(v)^{(t)}, \hat{b}^{(t)}\right)\right|<$ Tol.

To select the hyper-parameters of sparse SVQR using IRWLS, instead of the cross validation (CV) function used in SVQR using QP with check function (1), we consider CV function as follws:

$$
\begin{equation*}
C V(\boldsymbol{\lambda})=\frac{1}{n} \sum_{i=1}^{n} w t_{i}(\theta)\left(y_{i}-\hat{q}_{\theta}^{(-i)}\left(\boldsymbol{x}_{i} \mid \boldsymbol{\lambda}\right)\right)^{2} \tag{18}
\end{equation*}
$$

where $\boldsymbol{\lambda}$ is a set of hyper-parameters and $\hat{q}_{\theta}^{(-i)}\left(\boldsymbol{x}_{i} \mid \boldsymbol{\lambda}\right)$ is the $\theta$ th quantile regression function estimated without $i$ th observation. Since for each candidates of hyper-parameters, $\hat{q}_{\theta}^{(-i)}\left(\boldsymbol{x}_{i} \mid \boldsymbol{\lambda}\right)$ for $i=1, \cdots, n$, should be evaluated, selecting hyper-parameters using CV function is computationally formidable. By using leaving-out-one lemma [18] the ordinary cross validation (OCV) function can be obtained as

$$
\begin{align*}
\operatorname{OCV}(\boldsymbol{\lambda}) & =\frac{1}{n} \sum_{i=1}^{n} w t_{i}(\theta)\left(\frac{y_{i}-\hat{q}_{\theta}\left(\boldsymbol{x}_{i} \mid \boldsymbol{\lambda}\right)}{1-\frac{\partial \hat{q}_{\theta}\left(\boldsymbol{x}_{i} \mid \boldsymbol{\lambda}\right)}{\partial y_{i}}}\right)^{2} \\
& =\frac{1}{n} \sum_{i=1}^{n} w t_{i}(\theta)\left(\frac{y_{i}-\hat{q}_{\theta}\left(\boldsymbol{x}_{i} \mid \boldsymbol{\lambda}\right)}{1-h_{i i}}\right)^{2} \tag{19}
\end{align*}
$$

where $H$ is the hat matrix such that $\hat{q}_{\theta}(\boldsymbol{x} \mid \boldsymbol{\lambda})=H \boldsymbol{y}$ with the $(i, j)$ th element $h_{i j}=\partial \hat{q}_{\theta}\left(\boldsymbol{x}_{i}\right) / \partial y_{j}$. Here the hat matrix is given as

$$
H=(K, \mathbf{1})\left(\begin{array}{cc}
K W K+\gamma V & K W \mathbf{1} \\
\mathbf{1}^{\prime} W K & \mathbf{1}^{\prime} W \mathbf{1}
\end{array}\right)^{-1} \quad\binom{K W}{\mathbf{1}^{\prime} W}
$$

where $W$ and $V$ are the final estimates. Replacing $h_{i i}$ by their average $\operatorname{tr}(H) / n$, the generalized cross validation (GCV) function can be obtained as

$$
\begin{equation*}
\operatorname{GCV}(\lambda)=\frac{n \sum_{i=1}^{n} w t_{i}(\theta)\left(y_{i}-\hat{q}_{\theta}\left(x_{i} \mid \lambda\right)\right)^{2}}{(n-\operatorname{tr}(H))^{2}} \tag{20}
\end{equation*}
$$

## Numerical Studies

In this section, we illustrate the performances of the proposed sparse SVQR using IRWLS procedure through the simulated examples by comparing SVQR using QP (SVQR_QP). RBF kernel is utilized in numerical studies.
Example 1. An objective of microarray analysis is to identify genes differently expressed under two distinct experimental conditions. This task is complicated since the noisiness of data exists and the large number of genes are examined. Sohn, Kim, Hwang, Lee and Shim [19] applied SVQR on cDNA microarray experiments to identify genes with different expression levels between two types of samples and showed SVQR performed well when error variability for each gene was heterogeneous in intensity ranges. We generate the artificial data set of microarrays analysis similar to $[19,20]$ by following steps:

1. Generate the true expression signal from an exponential distribution with mean 3000. Red (R) and green (G) channel intensities for each gene are generated from a normal distribution with mean of true expression signal and standard deviation of $15 \%$ of true expression signal, respectively.
2. Set $10 \%$ of genes to be over and under expressed. The selected genes are converted to $\mathrm{R} \times \sqrt{t}$ and $\mathrm{G} / \sqrt{t}$, where $t=10^{g}$ or $10^{-g}$ with same probability and $g$ is generated from Beta distribution with shape parameters 1.7 and 4.8.
3. Transform intensities nonlinear pattern with $f_{G}(z)=z(1-$ $\left.\exp \left(-z 100^{0.9}\right)\right)^{-0.9}$ and $f_{R}(z)=z\left(1-\exp \left(-z 100^{0.7}\right)\right)^{-0.7}$.
We generate a training data set and a test data set of 500 genes with 50 differently expressed genes, respectively. The standardized $\log$ ratio $M=\log _{2}(R / G)$ and the average $\log$ intensity $A=\log _{2} \sqrt{R G}$ are used as response and input variable, respectively. We repeated the above procedure 100 times. We selected the optimal values of penalty and kernel parameter using GACV function (8) and GCV function (20) for $\theta=0.975$ and $\theta=0.025$, respectively, for SVQR_OP and the proposed SVQR. Fig. 1 shows the predicted quantile regression functions by SVQR_QP (dotted lines) and by the proposed SVQR (solid

Table 1. Average number of genes selected, average number of true genes selected, average sensitivity and average specificity for each pattern (standard error in parenthesis).

|  | No of genes selected | No of true genes selected | Prop of true genes selected | Sensitivity | Specificity |
| :---: | :---: | :---: | :---: | :---: | :---: |
| QP | 39.2400 | 22.5400 | 0.5817 | 0.4508 | 0.9499 |
|  | $(0.7108)$ | $(0.3686)$ | $(0.0083)$ | $(0.0074)$ | $(0.0008)$ |
| Proposed | 36.8400 | 22.4400 | 0.6203 | 0.4488 | 0.9501 |
|  | $(0.7349)$ | $(0.3817)$ | $(0.0101)$ | $(0.0076)$ | $(0.0008)$ |

Table 2. CPU time of training the quantile model using $S V Q R \_Q P$ and the proposed $S V Q R$ for $\theta=0.1,0.5,0.9$.

| $\theta$ | n | QP | Proposed | $\theta$ | n | QP | Proposed |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0.1 | 100 | 4.9 | 0.9 | 0.1 | 500 | 56.4 |  |
|  | 300 | 48.7 | 3.6 |  | 700 | 501.1 |  |
| 0.5 | 100 | 3.2 | 0.9 | 0.5 | 61.8 |  |  |
|  | 300 | 48.3 | 1.0 |  | 400 | 175.6 |  |
| 0.9 | 100 | 3.2 | 5.1 | 0.9 | 500 | 12.1 |  |
|  | 300 | 48.3 |  |  | 700 | 49.3 | 11.2 |



Fig. 1. Predicted quantile regression functions by SVQR_QP (dotted lines) and by the proposed SVQR (solid lines), superimposed scatter plots of the log ratio versus the average log intensity.
lines), superimposed scatter plots of $M$ versus $A$ for one of 100 test data sets, where + denotes differently expressed genes. We compared both average sensitivity and specificity. As shown in Table 1, both methods provide almost same performance on sensitivity and specificity. But the proposed SVQR has a higher proportion of true genes selected than SVQR_QP though the proposed SVQR has almost same number of genes selected as SVQR_QP.
Example 2. We generate a data set in a similar manner to [21]. The univariate input variables $x$ 's are drawn from a uniform distribution $U(0, \pi)$ and the corresponding responses $y$ are drawn from a univariate normal distribution with mean and
variance that vary smoothly with $x$ as follows,

$$
y \sim N\left(\sin \left(\frac{5 x}{2}\right) \sin \left(\frac{3 x}{2}\right), \frac{1}{100}+\frac{1}{4}\left[1-\sin \left(\frac{5 x}{2}\right)\right]^{2}\right) .
$$

We compare CPU-times of the proposed with that of QP computed by the built-in function of MATLAB. Here ( $\gamma, C, \sigma^{2}$ ) are fixed as (200, 200, 1.5). Table 2 shows CPU-times in seconds of both methods (run MATLAB R2010a over Pentium IV at 4.0 $\mathrm{GHz})$ on data sets with different sample sizes. From Table 2 we can see that the computation of the proposed SVQR is much faster than that of SVQR_QP, which implies that the proposed SVQR is appropriate method for the large data.

## Conclusions

In this paper, we dealt with estimating the quantile regression function by sparse SVQR using IRWLS procedure, where the weighted quadratic loss function and $l_{1}$ norm penalty term are used instead of the check function and $l_{2}$ norm penalty term in SVQR_QP. Through the examples we showed that the proposed method derives the satisfying results - good prediction ability and fast computation. We also found that sparse SVQR using IRWLS procedure has an advantage other than SVQR_ QP, which is applicable to the large data set frequently found in microarray data analysis.

## Acknowledgements

This research was supported by Basic Science Research Pro-
gram through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012000646).

## References

1. Koenker R, Bassett G. Regression quantile. Econometrica 1978; 46:33-50.
2. Koenker R, Hallock KF. Quantile regression. J Econ Perspect 2001;40:122-142.
3. Yu K, Lu Z, Stander J. Quantile regression: applications and current research area. Statistician 2003;52:331-350.
4. Koenker R. Quantile regression. Cambridge: Cambridge University Press; 2005.
5. Shim J, Hwang C. Support vector censored quantile regression under random censoring. Comput Stat Data An 2009;53:912917.
6. Vapnik VN. The nature of statistical learning theory. New York: Springer; 1995.
7. Vapnik VN. Statistical learning theory. New York: John Wiley; 1998.
8. Smola A, Scholkopf B. On a kernel-based method for pattern recognition, regression, approximation and operator inversion. Algorithmica 1998;22:211-231.
9. Tipping ME. Sparse Bayesian learning and the relevance vector machine. J Mach Learn Res 2001;1:211-244.
10. Hwang C, Shim J. A simple quantile regression via support vector machine. Lect Notes Comput Sc 2005;3610:512-520.
11. Takeuchi I, Le QV, Sears TD, Smola AJ. Nonparametric quantile estimation. J Mach Learn Res 2006;7:1231-1264.
12. Mercer J. Functions of positive and negative type and their connection with theory of integral equations. Philos T Roy Soc A 1909;209:415-446.
13. Kuhn HW, Tucker AW. Nonlinear programming. Proceedings of 2nd Berkeley Symposium: 481-492. Berkeley; July. 1951.
14. Yuan M. GACV for Quantile smoothing splines. Comput Stat Data An 2006;50:813-829.
15. Li Y, Liu Y, Zhu J. Quantile regression in reproducing kernel Hilbert spaces. J Am Stat Assoc 2007;102:255-268.
16. Williams PM. Bayesian regularization and pruning using a Laplace prior. Neural Comput 1995;7:117-143.
17. Krishnapuram B, Carlin L, Figueiredo MAT, Hartermink AJ. Sparse multinomial logistic regression: fast algorithms and generalization bounds. IEEE T Pattern Anal 2005;27:957-968.
18. Craven P, Wahba G. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. Numer Math 1979;31:377-403.
19. Sohn I, Kim S, Hwang C, Lee JW, Shim J. Support vector machine quantile regression for detecting differently expressed genes in microarray analysis. Method Inform Med 2008;47:459467.
20. Balagurunathan Y, Dougherty E, Chen Y, Bittner M, Trent J. Simulation of cDNA microarrays via a parameterized random signal model. J Biomed Opt 2002;7:507-523.
21. Cawley GC, Talbot NLC, Foxall RJ, Dorling SR, Mandic DP. Heteroscedastic kernel ridge regression. Neurocomputing 2004; 57:105-124.

[^0]:    * Correspondence should be addressed to Dr. Jooyong Shim, Institute of Statistical Information, Department of Data Science, Inje University, Gimhae 621-749, Korea. Tel: +82-55-320-3290, Fax: +82-55-320-3587, E-mail: ds1631@hanmail.net

